# ELIMINATING DESIGN ALTERNATIVE UNDER INTERVAL-BASED UNCERTAINTY

A Thesis
Presented to
The Academic Faculty

by

Steven Joseph Rekuc

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2005

# ELIMINATING DESIGN ALTERNATIVES UNDER INTERVAL-BASED UNCERTAINTY

Approved by:

Dr. Chris Paredis, Advisor
School of Mechanical Engineering
*Georgia Institute of Technology*

Dr. Bert Bras
School of Mechanical Engineering
*Georgia Institute of Technology*

Dr. Leon McGinnis
School of Industrial and Systems Engineering
*Georgia Institute of Technology*

Date Approved: July 12, 2005

To my parents: for their love, support, and investment in my growth.

# ACKNOWLEDGEMENTS

A Master's Thesis is a project of immense scope that can seem insurmountable, and my thesis would have been, if it were not for the support of my colleagues, friends, and family.

I would like to thank my advisor, Dr. Chris Paredis, for his continued support and instruction. Chris gave me the opportunity to grow as one of his students and an insight into the world from a researcher's perspective. Without Chris' continual effort, this thesis would not be possible and I would not have grown nearly as much.

I would also like to thank the members of my committee, Dr. Bert Bras and Dr. Leon McGinnis for taking the time out of their busy schedules to provide valuable critique of this thesis.

The members of the Systems Realization Laboratory have provided immense support throughout my graduate learning experience that has enabled my immense intellectual and personal growth.

I would like to thank my friends that have made my time in Atlanta enjoyable and the hardships of a thesis that much easier.

Most of all, I would like to thank my family, who have provided the love and encouragement throughout my graduate experience and my life.

# TABLE OF CONTENTS

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

# SUMMARY

Typically, design is approached as a sequence of decisions in which designers select what they believe to be the best alternative in each decision. While this approach can be used to arrive at a final solution quickly, it is unlikely to result in the most-preferred solution.  The reason for this is that all the decisions in the design process are coupled. To determine the most preferred alternative in the current decision, the designer would need to know the outcomes of all future decisions, information that is currently unavailable or indeterminate. Since the designer cannot select a single alternative because of this indeterminate (interval-based) uncertainty, a set-based design approach is introduced.  The approach is motivated by the engineering practices at Toyota and is based on the structure of the Branch and Bound Algorithm. Instead of selecting a single design alternative that is perceived as being the most preferred at the time of the decision, the proposed set-based design approach eliminates dominated design alternatives: rather than selecting the best, eliminate the worst. Starting from a large initial design space, the approach sequentially reduces the set of non-dominated design alternatives until no further reduction is possible – the remaining set cannot be rationally differentiated based on the available information. A single alternative is then selected from the remaining set of non-dominated designs.

In this thesis, the focus is on the elimination step of the set-based design method: A criterion for rational elimination under interval-based uncertainty is derived. To be efficient, the criterion takes into account shared uncertainty – uncertainty shared between design alternatives. In taking this uncertainty into account, one is able to eliminate significantly more design alternatives, improving the efficiency of the set-based design approach. Additionally, the criterion uses a detailed reference design to allow more elimination of inferior design sets without evaluating each alternative in that set. The effectiveness of this elimination is demonstrated in two examples: a beam design and a gearbox design.

# CHAPTER 1

# THE CHALLENGE OF DESIGN UNDER UNCERTAINTY

Engineering designers face many challenges in the design process, and one of the more significant challenges is the uncertainty in the design process. Because a designer faces significant uncertainty in engineer design, the designer must make design decisions without knowing the exact outcome. This problem is only exacerbated by the typical approach to design decisions. The designer needs an approach to control uncertainty and make confident decisions based on uncertain information. I present the seed of such an approach.

## *1.1 The Typical Engineering Design Approach*

From the decision-based design perspective, designing can be viewed as making a sequence of decisions with the goal to arrive at the most-preferred design (Tribus 1969; Sage 1977; Hazelrigg 1998) (Mistree, Smith et al. 1990). This perspective serves as a good model for understanding the design process and has sparked much academic work (Thurston 1999; Gu, Renaud et al. 2000; Chen 2001; Chen 2003), but these decisions-based design approaches share a common problem with the typical approach to design decisions.

The problem originates from the designers approaching each design decision as if it were the only decision in the process. Designers often decide on a single alternative that they believe to be the best choice. I refer to this practice as making a *point decision*. In a single decision, one could select the most preferred alternative; one could even make a point decision under some uncertain conditions, in which the design that is most likely to be the most preferred design is selected. Since this point-decision approach works for a single decision, designers apply it to the sequence of decisions in design.

Although making a sequence of point decisions can be used to arrive at a final solution quickly, it is unlikely to result in the most-preferred design. This is because all the decisions in the design process are coupled: the decision the designer makes in one decision is dependant on the result of the other decisions. To determine the alternative in the first decision that leads to the most preferred design, the designer would need to know the outcomes of all future decisions. Since the designer does not know the outcomes of all future decisions, the designer could not select the single alternative in each decision that leads to the most preferred design.

For example, if one is designing an automobile, the specific design variables one chooses for a transmission are dependent on the engine one chooses. One could not know the specific transmission that results in the most preferred automobile without knowing the specific engine. If one chose a specific transmission without significant information about the engine to be used then that choice likely would not lead to the most preferred automobile.

A designer cannot find the most preferred solution by a sequence of point decisions because of uncertainty about the outcomes of the future decisions. This uncertainty is typically represented by an interval, where designers represent the range of possible choices in the future decisions. The interval representation has also been used to represent experimental error, bound the uncertainty in computational models, and determine the effects of round-off error (Ferson and Ginzburg 1996; Kearfott and Kreinovich 1996; Hayes 2003). This representation is a contrast to the probability distribution used in many uncertainty applications. Both of these representations are reviewed in Section 2.1, where the differences are examined.

The difference in these uncertainty representations is important in the decision process. While many methods have been developed for making decisions under probability-based uncertainty (Luce and Raiffa 1957; Keeney and Raiffa 1993; Triantaphyllou 2000; Fernandez, Seepersad et al. 2001; Stirling 2003), there is a need for making decisions under interval-based uncertainty. In this thesis, I address this problem by creating a criterion for deciding under interval-based uncertainty.

## *1.2 My Approach to Design*

In considering interval-based uncertainty, I approach the design problem from a different perspective. I still decouple the problem into a sequence of decisions, but because the most preferred solution cannot be determined via a series of point decisions, I do not force the decision maker to make *point decisions*. Instead, I propose that the designer decide on the *set* of design alternatives. In this *set-based design* approach, the designer decides on a set of possible solutions, eliminating the alternatives or values from the set that are guaranteed not to lead to the most preferred design based on and consistent with the currently available information and knowledge. The focus is thus on elimination rather than selection.

In this thesis, I address the issue of how one should eliminate. Since this elimination approach is motivated by interval-based uncertainty in future decisions, I focus on how one should eliminate under interval-based uncertainty in my research question:

> **Question:** *Under conditions of interval-based uncertainty, how should one eliminate design alternatives?*

The goal of the designer is to find the most preferred design, thus the designer should eliminate all designs that cannot be the most preferred design. This is rational elimination, and it serves as the underlying principle pointed out in my hypothesis:

**Hypothesis:** *One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design and by accounting for shared uncertainty.*

The hypothesis has three main points. First, one should eliminate design alternatives rationally; this idea has already been introduced to the reader. The second aspect in the hypothesis relates to comparing the design alternatives to a specific reference design for elimination, and the third aspect accounts for shared uncertainty. These aspects of the hypothesis are explained in full in Chapter 4.

Elimination is a crucial step in the set-based design process and the focus of this thesis; however, to examine the elimination step, a method for managing the set-based design process is needed. Since no formal set-based design method exists, I propose the concept for a design method based on another set-based process of elimination: the Branch and Bound Algorithm (B&B). The details of this concept are given in Chapter 3, along with requirements for this approach to be successful. Although I present the concept of a B&B approach to design, this is not the contribution of this thesis; instead, this concept is the context in which to view the elimination method that is developed. The elimination method could be used in any formal set-based design process.

With my perspective defined, the research questions and hypothesis posed, the focus is shifted to detailing and validating the research in the chapters that follow. To guide the reader through the chapters that follow the next section presents a roadmap for the rest of this thesis.

## *1.3 Guide to My Thesis and Validation*

The rest of this thesis will be dedicated to developing and validating the research with respect to the research questions and hypotheses. Since this work revolves around the research questions and hypothesis those questions and hypothesis are reiterated in Table 1.

**Table 1: Research Question and Hypothesis**

| | |
|---|---|
| Question | Under conditions of interval-based uncertainty, how should one eliminate designs? |
| Hypothesis | One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design and by accounting for shared uncertainty. |

To validate this hypothesis, a validation strategy based is necessary. For this, the validation square, as developed by Pederson and coauthors (Pederson, Emblemsvag et al. 2000), is used. The validation square is specifically for validating design methods; as this validation procedure demands one establish the theoretical structural integrity of a design method as well as the relevance of that method. These two aspects in validating a design method are broken down into four procedures given in each quadrant of the validation square, as shown in *Figure 1*. In validating a method, one begins at the upper left corner; in this quadrant one establishes the theoretical validity of the method with the central issue being: does this method theoretically produce the desired result? After establishing the theoretical validity one should move to the next quadrant which begins the focus on the relevance or usefulness of the method.

In validating a method, one begins at the upper left corner; in this quadrant one establishes the theoretical validity of the method with the central issue being: does this method theoretically produce the desired result? After establishing the theoretical validity one should move to the next quadrant which begins the focus on the relevance or usefulness of the method.

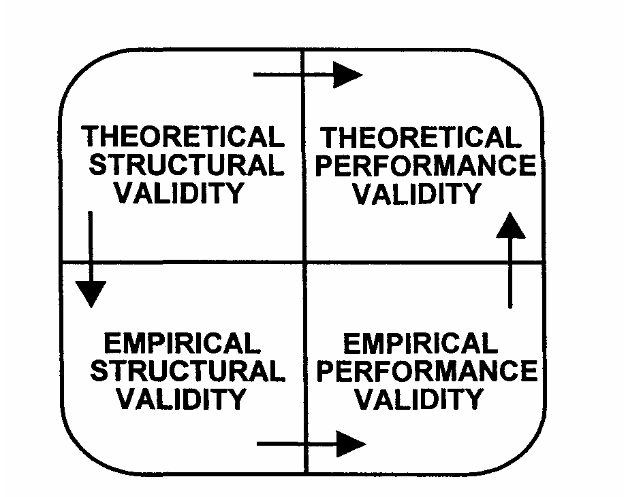



*Figure 1: The Validation Square (Pederson, Emblemsvag et al. 2000)*

Establishing the relevance of a method is a much more trying process then just establishing the theoretical structural validity and requires the remainder of the validation square beginning with second procedure in the bottom left. In this quadrant, one establishes the validity of the example problem(s) to test the relevance of the design method. The example problems are then used to test to the performance of the design method; this is corresponds to the third quadrant of the validation square. The final quadrant of the validation square is requires a 'leap of faith' to accept that based on the theoretical structure and the empirical results the design method can be accepted as generally valid.

To obtain validity the design method proposed in this thesis uses the validation strategy from the validation square. The quadrants of the validation square are explained as they relate to each chapter of the thesis in Table 2. This table may leave the reader at a loss for understanding the organization of this thesis. To this end, Figure 2 is presented to act as a roadmap for this thesis, detailing the importance of each chapter in the thesis. This roadmap has been modified from the work of Seepersad (Seepersad 2001). With the research problem stated and validation strategy in place, the reader's attention is shifted to filling out the details of the research and validation. This begins with a review of previous work in Chapter 2.

**Table 2: Strategy for Validation**

| Quadrant of Validation Square | Thesis Chapter/Section | Aspect of Validation |
|---|---|---|
| Theoretical Structural Validity (1) | Chapter 2 | Literature Search to establish theoretical basis for proposed method |
| | Chapter 4: Section *4*.2 | Theoretical structural soundness of rational decision-making |
| Empirical Structural Validity (2) | Chapter 4: Section 4.3 | Soundness of applying common uncertainty to elimination |
| | Chapter 5: Section 5.2 | Example's capability to test elimination method is established |
| Empirical Performance Validity (3) | Chapter 5: Section 5.5 | Example is evaluated to determine if design method is useful |
| | Chapter 5: Section 5.6 | Example decisions examined to verify performance of elimination method |
| Theoretical Performance Validity (4) | Chapter 6: Section 6.1 | Hypothesis is revisited and examined for validity to make 'Leap of Faith' |

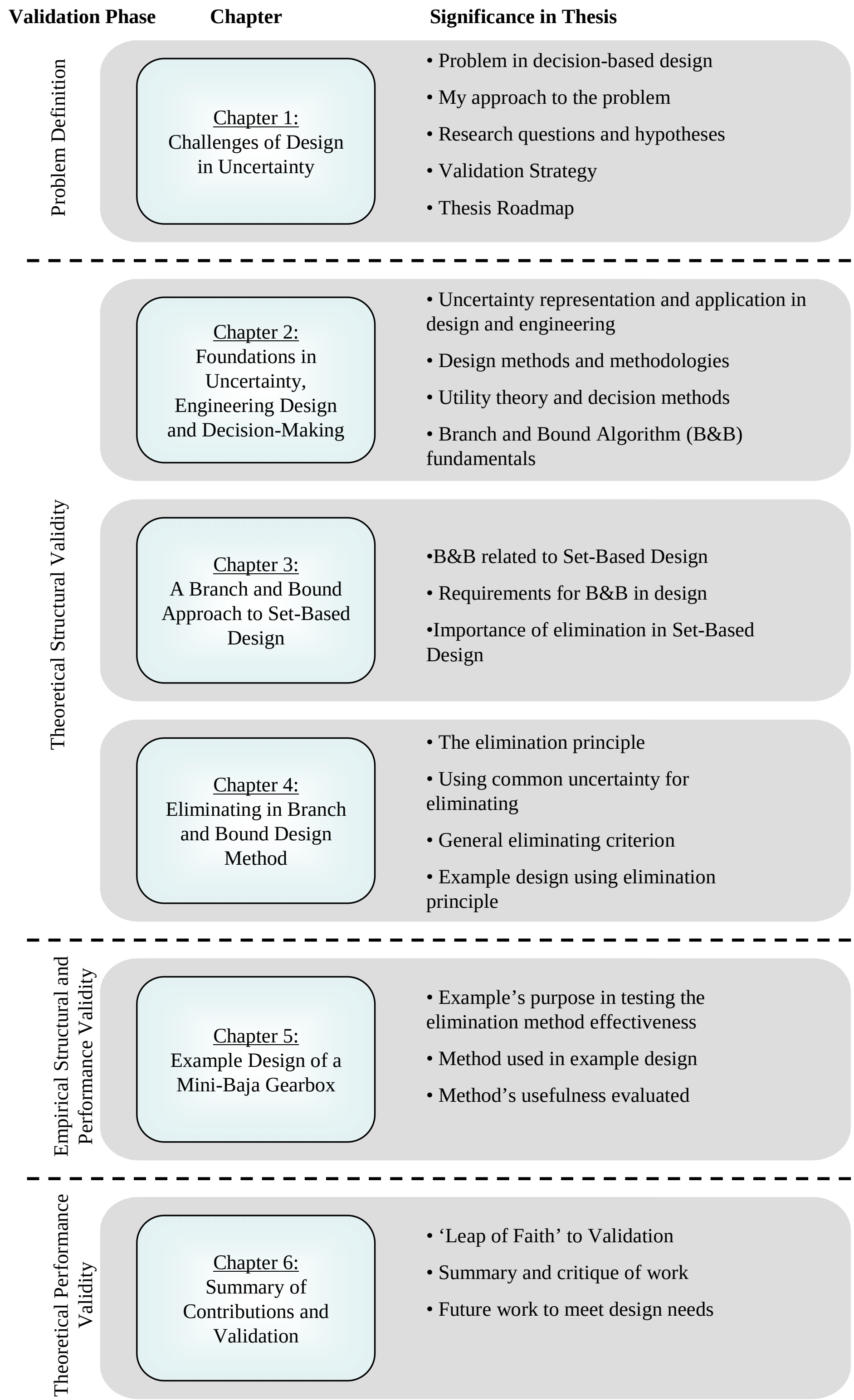



*Figure 2: Thesis Roadmap*

# CHAPTER 2

# FOUNDATIONS IN UNCERTAINTY, ENGINEERING DESIGN AND DECISION-MAKING

As explained in Chapter 1, uncertainty has a profound impact on the design process, thus this review focuses on uncertainty and its role in design and decisions methods. The review begins with the nature of uncertainty and its different representations in Section 2.1. This understanding of uncertainty is then applied to review design approaches for their ability to handle uncertainty in Section 2.2. Section 2.3 presents the Branch and Bound (B&B) algorithms as a concept for set-based design under uncertainty that the elimination approach could be viewed in. Since elimination is similar to decision making, formal decision making methods are reviewed in Section 2.4. Specifically, utility theory is reviewed as the basis of rationality that will be later extended to interval-based uncertainty. Section 2.5 summarizes the work reviewed in this chapter, pointing out both the relevant contributions and the opportunities for improvement.

## *2.1 Uncertainty in Engineering Design*

Uncertainty is ubiquitous in engineering design. First, systems are designed to perform in the future, which is uncertain. Second, design performance is predicted by using a model; either a prototype or a computational model. Because models are, by definition, an abstraction of reality, they cannot perfectly predict reality and their results contain some uncertainty. Lastly, the design process often is decomposed into a series of decisions and the designer addresses one decision at a time. (Sage 1977; Mistree, Smith et al. 1990; Hazelrigg 1998; Thurston 1999; Chen 2001; Chen 2003) This introduces uncertainty because in a sequence of decisions, the outcome of the future decisions, which affect performance, often are not known. With all of these sources of uncertainty in the design process, a means of handling this uncertainty is necessary.

One common approach to account for uncertainty is to use safety factors in design. The idea behind safety factors is that the designer accounts for ignored uncertainty by multiplying some aspect of their calculation by a safety factor to make the design more conservative. While this approach has been shown to have some mathematical basis (Elishakoff 2004), the approach is severely flawed: If one does not know the uncertainty, then one cannot know whether the safety factor is large enough to account for this uncertainty. Thus, this approach results in designers using safety factors that are either too large, resulting in over-designed systems, or too small, resulting in design failure. To avoid this, one needs to recognize and appropriately handle uncertainty in design.

In this thesis, I present a method for eliminating designs under interval-based uncertainty. To understand when this method applies, one needs to understand when the interval representation of uncertainty is appropriate. In this section, I establish an understanding of uncertainty. Specifically, this section covers the types of uncertainty in design, modeling of the uncertainty, as well as the appropriateness and value of those models. Much of what one does with uncertainty depends on how one views uncertainty, thus to begin this investigation, the nature of uncertainty is examined in Section 2.1.1. Then in Section 2.1.2, the representation of uncertainty is introduces before the interval representation is presented in Section 2.1.3 and the probability distribution in Section 2.1.4.

### 2.1.1 Nature of Uncertainty

The term uncertainty is used to describe incomplete information. Researchers have recognized that this incomplete information arises from one of two basic phenomena: naturally random behavior, called aleatory uncertainty, or a lack of knowledge, called epistemic uncertainty (Antonsson and Otto 1995; Parry 1996; Oberkampf, DeLand et al. 2002).

Aleatory uncertainty is due to naturally random (stochastic) behavior. Aleatory uncertainty is also known as variability, stochastic uncertainty, objective uncertainty, and irreducible uncertainty (Oberkampf, DeLand et al. 2002). An example of this uncertainty is the roll of a die. The roll is random – the actual outcome of a particular roll is uncertain – only the probability of each outcome is known. Examples of this uncertainty in engineering range from the distribution of material properties to variability in machine operation. This uncertainty is inherent in the system and cannot be reduced.

Conversely, epistemic uncertainty is due to a lack of knowledge or a lack of application of knowledge (Parry 1996; Oberkampf, DeLand et al. 2002). Epistemic uncertainty is also called imprecision (Antonsson and Otto 1995) or reducible uncertainty, and can be reduced or eliminated through the discovery of new information or knowledge. For example, one may initially model the position of a falling object without considering aerodynamic drag. Omitting the drag component introduces a systematic—but unknown—error in the model. That error could be eliminated from the model by including the drag. Since by definition, all models are abstractions of reality, some epistemic uncertainty will always remain in any model used in design. Additionally, epistemic uncertainty is introduced by the process of design; a designer does not know the result of future decisions, but will eventually eliminate that uncertainty by making those decisions.

Philosophically, the difference between aleatory and epistemic uncertainty may not always be clear. Some authors have argued that they are fundamentally indistinguishable (Winkler 1996). Sometimes, it is difficult to distinguish between precise examples of different types of uncertainty. One can never know for sure whether more analysis (and the resulting knowledge created) might reduce what was once thought to be intrinsic variability to something that can be deterministically predicted (Berleant, Cheong et al. 2003). For example, the roll of a die is viewed as inherently random, as stated before; however, if every detail of the roll was modeled with very accurate physical models and the initial conditions were accurately known, then maybe the roll of the die could be predicted deterministically.

Despite such philosophical issues, when the context of decisions is limited to engineering design, it is valuable to make the distinction between aleatory and epistemic uncertainty. First, the distinction allows one to determine what uncertainty can be reduced. One can reduce epistemic uncertainty, by uncovering and applying additional knowledge, at a far lower cost than one could reduce aleatory uncertainty (Aughenbaugh and Paredis 2005).

Second, from my perspective, the distinction results in different mathematical representations. Aleatory uncertainty, since it is random with known probabilities, should be modeled as a probability distribution, whereas epistemic uncertainty, which is not a product of an underlying random event, is better represented by an interval that bounds the possible outcomes. To understand fully my reasons for choosing these representations, one must understand the theory behind these representations; this is presented in the next section.

### 2.1.2 Uncertainty Representation Formalisms

Through their experience and education, engineers have internalized significant knowledge and information. Such knowledge is often *implicit* and is available only to that particular engineer. In order for this knowledge or information to be used by others, it has to be made *explicit*. By expressing uncertain information explicitly in a mathematical formalism, that information is available for analysis and decision-making.

An uncertainty formalism is a mathematical model to represent uncertain information. Like all models, it is an abstraction of reality, thus the translation from reality into the formal model involves an unavoidable loss of information. This loss is offset in value by the benefit of making the information explicit and in a form with which can be computed – the resulting information is more useful, and thus valuable, than in its original state. Therefore, representing information in a formal manner should result in an increase in value of that information. The value of the information is a strong consideration in deciding what information should be represented in a formal manner.

In representing uncertainty, one strives to express as much information as possible about the unknown. The more information is expressed in the representation, the more information one can use to make decisions, thus the more valuable that representation. For example, if one has the data given in Figure 3 for values of an uncertain parameter, one could bound this uncertain parameter with an interval: [2, 8]. This would be correct, but the interval is overly conservative and limits one's ability to use this information to make a decision. One would have a more valuable representation using an interval [5, 7] instead, and avoid being overly-conservative.

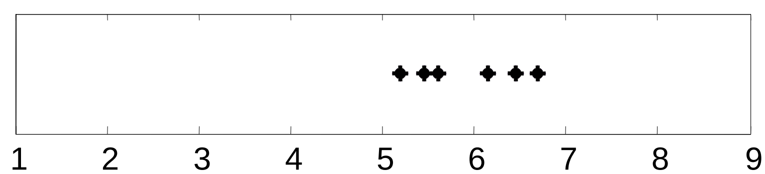



*Figure 3: Data of an uncertain parameter to be characterized with an uncertainty representation.*

Conversely, one does not want to over-state that information in a representation. In doing so, the representation could not be trusted in the decision process, and loses value. For example, if one has the data given in Figure 3 for values of an uncertain parameter, one could represent this uncertain parameter with the interval [5.5, 6.5]. However, this would be stating much tighter bounds than is supported by the known information. This representation could lead a designer to make a wrong decision and therefore is less valuable.

Thus, in using a formal uncertainty representation, one needs to be conservative without being overly-conservative. A valuable formalism accurately reflects what the designer knows and does not imply information that the designer doesn't have. This understanding is applied to the investigation of the two most widely applied representation formalisms in the next two sub-sections. First, the interval representation, which I've used in my elimination method, is presented in Section 2.1.3. Then the popular probability distribution is presented in Section 2.1.4.

### 2.1.3 Interval Analysis

Intervals have long been used to represent uncertainty. In one of the earliest of such uses, Archimedes inscribed and circumscribed polygons about a circle to obtain upper and lower bounds on $\pi$ (Berleant, Cheong et al. 2003). In a similar manner, the interval representation has been used throughout engineering to represent tolerance, model inaccuracy, or quantify other epistemic uncertainty. The interval representation is reviewed in this section. More specifically, an introduction to the interval representation is provided, the importance of this representation in engineering is shown, an example of interval representation is presented, and finally the advantage and limitations are pointed out.

In the interval representation, an uncertain parameter, $z$, is modeled with a closed interval, such as $Z = \left[\underline{z}, \overline{z}\right]$. This interval represents a set of all real numbers between the endpoints, $\underline{z}$ and $\overline{z}$, such that $Z = \{z \mid \underline{z} \le z \le \overline{z}\}$ (Moore 1966; Moore 1979; Kearfott and Kreinovich 1996; Hansen and Walster 2004). The meaning of Z is that the uncertain parameter, $z$, will *always* fall in that interval.

The interval representation is not limited to the closed intervals, which includes its endpoints; open intervals, which exclude their endpoints, could be used as well. The open interval, denoted by $Z = (\underline{z}, \overline{z})$, is used significantly less often in representing uncertainty because it is more difficult to compute with. Thus, closed intervals have been the chosen representation (Hansen and Walster 2004).

The intervals do not have to be represented by their endpoints, instead an interval such as $Z = [\underline{z}, \overline{z}]$ can be represented by a center point with the error bounds, such as: $Z = \tilde{z} \pm \varepsilon_z$, where $\tilde{z} = median(\underline{z}, \overline{z})$, $\varepsilon_z = \tilde{z} - \underline{z} = \overline{z} - \tilde{z}$. The center point and error bound representation is commonly applied in representing tolerances or scientific measurements, but is not typically used in computation.

**Representing Uncertainty with Intervals**

Although the interval representation is simple, it is not obvious how one obtains the *absolute* bounds for an interval. Quite simply, how does one assess the bounds for an interval? Interval Analysis literature focuses on the mathematics of computing with intervals and distinctly ignores this important point. This leaves one to look at the application of intervals to determine if a consistent method exists.

Although intervals have been used extensively in engineering and science communities, as surveyed by Kearfott (Kearfott and Kreinovich 1996), each of these applications does not assess the intervals with a systematic method. Instead, the researchers evaluate the bounds based on information, physical principles, or their knowledge (Kearfott and Kreinovich 1996; Kearfott 1997) (Moore 1966; Moore 1979) (Broadwater, Shaalan et al. 1994; Koyluoglu, Cakmak et al. 1995). While the ways researchers assess the intervals may not be based on a consistent, rigorous method, there is some consistency in the way that the intervals are assessed. I have condensed these into a basic criterion for practically assessing intervals.

The intervals need to be assessed practically because being strict requires that intervals include every possible outcome. Since, in most situations, one could never be certain that every possible outcome is contained, the bounds extend to infinity. This is not practical. Rather, to obtain a practical interval, researchers appear to adhere to the following criteria that I have abstracted:

**1. Intervals should be consistent with the information available.** Although different people may place different bounds given the same information, a rational person would place bounds that are in contradiction with the information they possess. The interval needs to be consistent with the information that is known. For example, if one had the information in Figure 3, an interval of [5.5, 6.5] would contradict the information, whereas bounds of [5, 7] and [4.5, 7.5] are both consistent with the information.

**2. Intervals should be consistent with the knowledge of those setting the bounds.** Researchers know the physical laws that govern the domain where they are placing those bounds. The knowledge that they have of the domain can be used to set the interval, but most importantly, that knowledge should not be in contradiction with the interval. One of the most classic examples of applying knowledge to set the interval bounds is in the application of the Taylor Series Expansion, in which a more complicated function is represented by a polynomial expansion using its derivatives (Chapra and Canale 1998). The expansion takes the form:

$$f\left(x_{i+1}\right)=f\left(x_i\right)+f'\left(x_i\right)h+\frac{f''\left(x_i\right)}{2!}h^2+\frac{f'''\left(x_i\right)}{3!}h^3+...+\frac{f^{(n)}\left(x_i\right)}{n!}h^n+R_n,$$

where $f\left(x_i\right)$ is the value of the function at $x_i$, $f\left(x_{i+1}\right)$ is the value at the new point $x_{i+1}$, and $R_n$ is the remainder term, given by the remaining higher-order terms that have not been evaluated:

$$R_n=\frac{f^{(n+1)}\left(\xi\right)}{\left(n+1\right)!}h^{n+1}$$

where $\xi$ is a value of x that lies somewhere between $x_i$ and $x_{i+1}$. This remainder term can be used to bound the uncertainty in the result when applying the Taylor Expansion. First, the maximum value for the remainder term is considered:

$$R_{n,\max}=\max_{\xi\in\left[x_i,x_{i+1}\right]}\left|\frac{f^{(n+1)}\left(\xi\right)}{\left(n+1\right)!}h^{n+1}\right|$$

The result is bounds on the value of the function as follows:

$$f\left(x_{i+1}\right)=f\left(x_i\right)+f'\left(x_i\right)h+\frac{f''\left(x_i\right)}{2!}h^2+\frac{f'''\left(x_i\right)}{3!}h^3+...+\frac{f^{(n)}\left(x_i\right)}{n!}h^n\pm R_{n,\max}$$

**3. Valuable for the given situation.** With the first two criteria researchers still could set bounds that appear excessive for the given knowledge and information of the system. However, this is not practiced because it is not valuable. Just as in the example, the bounds that are set in the interval are only as wide as they need to be for the given problem – that is the most valuable bounds. Consider the example before based on the data in Figure 1. In that situation, there is a chance that an extreme event could occur that would result in an event outside of [5, 7], but to consider such an extreme event would not be valuable in most cases.

These are not strict criteria, but they do provide a practical means for assessing the interval bounds that I will apply in my example problem in Chapter 5. With at least a practical criterion for establishing an interval, attention is now turned to what these intervals have been and should be used for.

## Applications of Intervals

Intervals are a versatile representation, as they can be used to represent both aleatory and epistemic uncertainty. For aleatory uncertainty, one can bound all possible random outcomes with an interval, or assess a confidence interval, in which a significant portion of the outcomes will be contained. Both of these are appropriate if the probability distribution of the uncertain parameter is not known. However, if the distribution is known, an interval cannot express this information. This lack of information in the representation is fine if there is no value in representing it, but information about the probability is often valuable in making decisions. Since one would like to avoid this loss in information, aleatory uncertainty typically is not represented by an interval.

The intervals are well-suited for representing epistemic uncertainty. Because epistemic uncertainty is due to a lack of knowledge, the uncertainty is systematic. That is, the true value could be predicted perfectly by applying the appropriate knowledge or information that has been omitted. Since the uncertainty is systematic, one can bound the impact of the omitted knowledge or information to establish an interval. Thus, the interval formalism lends itself to representing epistemic uncertainty (Hayes 2003).

Intervals have been used widely to represent epistemic uncertainty, finding use in many academic and professional communities (Kearfott and Kreinovich 1996; Kearfott 1997) (Moore 1966; Moore 1979) (Broadwater, Shaalan et al. 1994; Koyluoglu, Cakmak et al. 1995). Among these communities, analysis with the interval representation progressed in a formal manner as an interest of computer programmers and their colleagues in mathematics. These individuals noticed that in computers, numbers could not be represented exactly; the numbers had to be truncated; this truncation introduces error in computing (Moore 1966; Moore 1979; Kearfott and Kreinovich 1996; Hansen and Walster 2004). The errors can be determined based on where the number was truncated. Since the calculations are truncated at a specific bit, and there is no knowledge of the bits after that, an interval representation is a natural fit for this epistemic uncertainty.

For example, consider representing the number $\pi$ with a decimal number. Technically, this representation of $\pi$ should continue to an infinite number of digits. However, one cannot represent all of these digits, so the representation is often truncated after 4 decimal places with 3.1416 as the result. The truncated digits will not be larger than the last digit included. Thus, the interval bounds on the true value of pi would be $\pi = \left[3.1415, 3.1416\right]$ when using this decimal representation. This same concept is applied to bound the binary representation used by computers.

Those in the computer science community have applied the interval analysis to search and optimization algorithms, specifically Branch and Bound Algorithms. In these algorithms, the region being searched is divided, or branched, into sub-regions. For each sub-region, the objective function is bounded by an interval. These intervals are used to eliminate sub-regions and the process continues for the remaining regions. These algorithms allow one to locate the global optimum even in large search spaces in a relatively short time (Moore 1991; Hansen and Walster 2004). These Branch and Bound Algorithms can be adapted for many optimization cases, but are associated most with mixed integer and integer programming (Nemhauser, Rinnooy Kan et al. 1989; Belegundu and Chandrupatla 1999). More detailed information on Branch and Bound is provided in section 2.5.

Epistemic uncertainty is introduced in behavior modeling, as every model is, by definition, an abstraction of reality. When one abstracts to create a model, one intentionally leaves out some knowledge or information and introduces uncertainty into that model's results (Ferson and Ginzburg 1996; Wojtkiewicz, Eldred et al. 2001; Malak Jr. and Paredis 2004). In a sense, the model has been truncated, much like the representation of numbers. Just like the computer truncation, nothing is known of what has been truncated and the error that occurs in the model can be attributed to the truncation. Since one knows the relative impact of what has been truncated, one can estimate the error in the model to be on the order of the truncation. This leads to using intervals in representing model uncertainty. Characterizing behavioral models is an important application of intervals in uncertainty.

For an example of how model uncertainty can be computed and represented with an interval, consider a system that is defined by the quadratic equation:

$$y(x) = 0.95x^2 + 2x + 1 \tag{2.1}$$

where $y$ is the system behavior, $x$ is the input to the system, and $0.95, 2, 1$ are system parameters. The system behaves in this manner for $x \in [0,5]$. In modeling the system, a linear model of the system performance is used, as given in Equation 2.4.

$$f(x) = 6.5x - 1 \tag{2.2}$$

where $f$ is the modeled system behavior, $x$ is the input to the system, and $6.5, -1$ are model parameters.

The uncertainty in the simplified model is determined by bounding the difference between the true system performance and the simplified model over the region $x \in [0,5]$. These bounds are calculated in Equation (2.5):

$$
\begin{aligned}
E &= [\underline{\varepsilon}, \overline{\varepsilon}] = \left[ \min_{x \in X} \left( f(x) - y(x) \right), \max_{x \in X} \left( f(x) - y(x) \right) \right] \\
E &= [\underline{\varepsilon}, \overline{\varepsilon}] = [-3.33, 3.25]
\end{aligned}
\qquad (2.3)
$$

where $E$ represents the error bounds on the model, and $[\underline{\varepsilon}, \overline{\varepsilon}] = [-3.33, 3.25]$ are the bounds, respectively. These bounds are shown in Figure 4, with both the model and the true system performance. These bounds on the model would be determined in advance of applying the model, as part of the model characterization. Then in applying the model one would use the following equation:

$$
F(x) = \left[ \underline{f}(x), \overline{f}(x) \right] = 6.5x - 1 + [-3.33, 3.25]
$$

This result gives bounds on the behavior of the real system such that: $y(x) \in F(x)$ Thus, this provides conservative bounds on the real result. The error represented in this model reflects the deficiencies in the model to reproduce reality.

Intervals also have been used for representing experimental error (Broadwater, Shaalan et al. 1994; Koyluoglu, Cakmak et al. 1995; Kearfott and Kreinovich 1996; Ben-Haim 2001). Researchers recognize that uncertainty in scientific experimental measurement introduces error in experimental results.

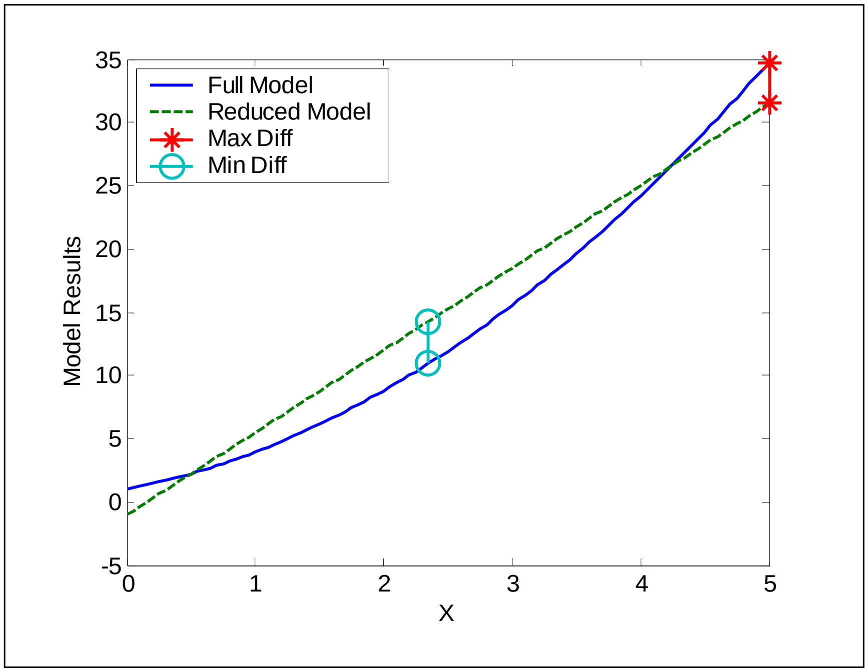



*Figure 4: Interval to represent uncertainty in a model.*

Since measurements from scientific experiments can only be determined to a certain level of precision, the interval representation was a natural choice, and the level of precision a natural indicator of the interval. This concept is shown in Figure 5, as one could determine that the object is greater than 3.5cm in length, but less than 4.0cm. Thus, the uncertainty in the length of the object leads to the interval representation.

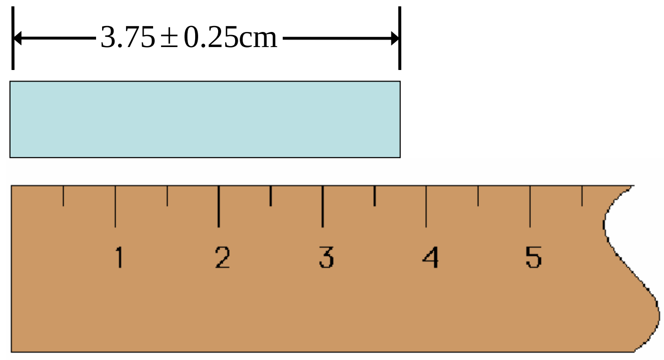



*Figure 5: Intervals to represent uncertainty in measurement.*

Additionally, intervals are used to represent the range of design variables being considered in design decisions. This uncertainty actually has a large impact on the design process, as pointed out in Chapter 1. Because design variables have a strong impact on design performance, uncertainty from these unspecified design variables results in large uncertainty about the design performance. This interval uncertainty from unspecified design variables makes it difficult to select a single alternative in a design decision; this is the motivation for the set-based approach to design.

### Basics of Computing with Intervals

Interval Analysis is based on the principle that it is better to bound the possible results, with an interval, rather than give a single answer with no indication as to how far the true answer could deviate from that point. In Interval Analysis, one computes with intervals such that the intervals are propagated through the calculations to accurately determine the uncertainty in the results. All inputs are bounded using intervals from the beginning. These intervals are carried through all computations, while rounding and other possible sources of error are incorporated into the resulting interval. The result is an absolute interval on the computation result (Moore 1966; Moore 1979; Kearfott and Kreinovich 1996; Hansen and Walster 2004).

At the foundation of interval analysis is interval arithmetic, which is an extension of the scalar arithmetic. Any binary interval arithmetic operation can be stated as:

$$X \bullet Y = \{x \bullet y \mid x \in X, y \in Y\} \tag{2.4}$$

where $X \bullet Y$ is the interval operation, and $x \bullet y$ is the corresponding operation on real numbers. Thus the result of $X \bullet Y$ must contain every possible result of $x \bullet y$ for all $x \in X$ and $y \in Y$. For addition with $X = [\underline{x}, \overline{x}]$ and $Y = [\underline{y}, \overline{y}]$, this results in the following for two intervals:

$$X + Y = [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \tag{2.5}$$

For subtraction the result is:

$$X - Y = [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \tag{2.6}$$

For multiplication, one needs to consider all possible multiplications between the endpoints of the intervals, and then select the minimum and maximum of these results. Thus, multiplication of two intervals becomes:

$$X * Y = \left[\min\left(\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\right), \max\left(\underline{x} * \underline{y}, \underline{x} * \overline{y}, \overline{x} * \underline{y}, \overline{x} * \overline{y}\right)\right] \tag{2.7}$$

Interval division, $X/Y$, can be computed in the same manner as long as $0 \notin Y$.

The interval operations presented here must be carried out through all computations in order to have results that accurately reflect the uncertainty. When calculations are performed on a computer, one must also be sure to round outward, extending the bounds on the interval to avoid introducing error by truncating. Interval Analysis extends far beyond this interval arithmetic and outward rounding; however, these basic interval operations provide a start toward a fundamental understanding.

In addition to having more complicated arithmetic operations, the interval operations are not all distributive; the order of the operations matters. For example, consider the expression:

$$[1,2]\times\left([-2,-1]+[2,3]\right) \tag{2.8}$$

If one performs the multiplication first and then the addition, the result is as follows:

$$\begin{gathered}[1,2]\times\left([-2,-1]+[2,3]\right)\\ [-4,-1]+[2,6]=[-2,5]\end{gathered} \tag{2.9}$$

This yields overly-conservative bounds, whereas if one performs the addition first and then the multiplication, the result is as follows:

$$\begin{gathered}[1,2]\times\left([-2,-1]+[2,3]\right)\\ [1,2]\times[0,2]=[0,4]\end{gathered} \tag{2.10}$$

Both of these intervals are correct, as they both bound the true solution. However, one would prefer $[0,4]$ over $[-2,5]$, as this is a much tighter bound on the solution. Thus, one needs to be concerned with the order of operations when computing with intervals.

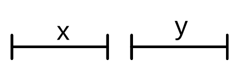
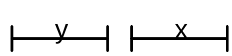
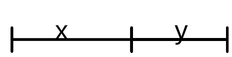
The order of operations is important with intervals because of dependence. Dependence refers to an instance of the same interval appearing in two different parts of an equation. When this occurs, the same intervals are computed with as if they were different instances. This dependence is seen in the example with interval $[1,2]$. In Equation (2.10), $[1,2]$ only appears once; however, in distributing the multiplication in Equation (2.9), the interval is actually used twice, as follows:

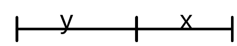
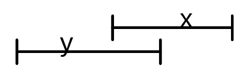
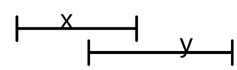
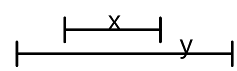
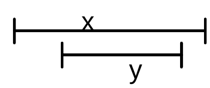
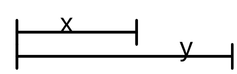
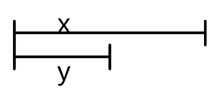
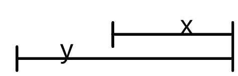
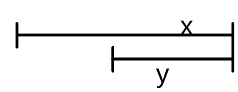
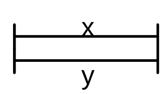
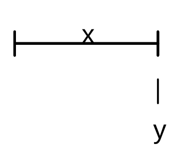
$$\begin{aligned}&[1,2]\times([-2,-1]+[2,3])\\&[1,2]\times[-2,-1]+[1,2]\times[2,3]\\&[-4,-1]+[2,6]=[-2,5]\end{aligned} \qquad (2.11)$$

This dependence causes the resulting interval to be overly-conservative. Whereas, the computations in Equation (2.10) perform the same operations (in a different order) with only one instance of the interval $[1,2]$. Because this computation avoids accounts for the dependence between the two instances of $[1,2]$, tighter bounds result on the outcome. To achieve tighter interval bounds, one should avoid duplicating intervals.

In addition to being complicated to compute with, intervals also are difficult to compare. Rather than have 3 basic relations, as there are with scalars, comparing intervals actually results in 18 different relations (Hayes 2003). These relations are given in Table 3. Each of these relations is composed of 4 different scalar relations that relate the bounds of the intervals being compared. The order of these relations is best explained in Table 4.

**Table 3. Interval Relations**

| **Relation** | **Constraint** | **Meaning** |
|---|---|---|
| X <<<< Y | $\overline{X} < \underline{Y}$ |  |
| X >>>> Y | $\overline{Y} < \underline{X}$ |  |
| X <<<= Y | $\overline{X} = \underline{Y}$ |  |

| | | |
|---|---|---|
| X =>>> Y | $\overline{Y} = \underline{X}$ |   |
| X <>>> Y | $\underline{Y} < \underline{X}$ and $\underline{X} < \overline{Y} < \overline{X}$ |   |
| X <<<> Y | $\underline{X} < \underline{Y}$ and $\underline{Y} < \overline{X} < \overline{Y}$ |   |
| X <><> Y | $\underline{Y} < \underline{X}$ and $\overline{X} < \overline{Y}$ |   |
| X <<>> Y | $\underline{X} < \underline{Y}$ and $\overline{Y} < \overline{X}$ |   |
| X <=<> Y | $\underline{X} = \underline{Y}$ and $\overline{X} < \overline{Y}$ |   |
| X <=>> Y | $\underline{Y} = \underline{X}$ and $\overline{Y} < \overline{X}$ |   |
| X <>=> Y | $\underline{Y} < \underline{X}$ and $\overline{X} = \overline{Y}$ |   |
| X <<=> Y | $\underline{X} < \underline{Y}$ and $\overline{Y} = \overline{X}$ |   |
| X <= => Y | $\underline{Y} = \underline{X}$ and $\overline{Y} = \overline{X}$ |   |
| X = = = =Y | $\underline{X} = \overline{X} = \underline{Y} = \overline{Y}$ |   |
| X < < = =Y | $\overline{X} = \underline{Y} = \overline{Y}$ |   |

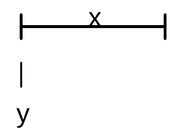
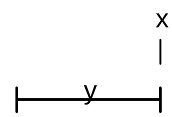
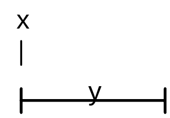
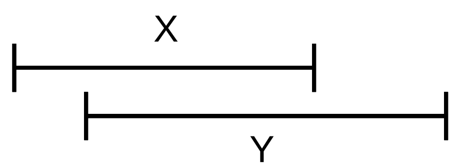
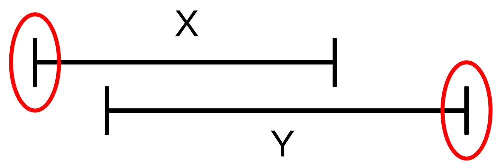
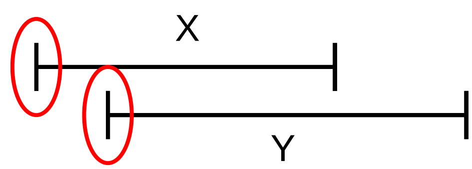
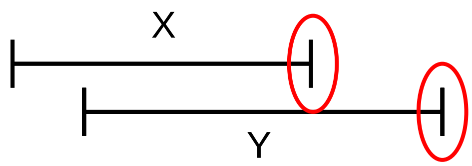
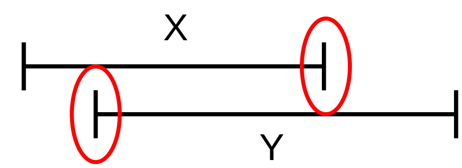
| | | |
|---|---|---|
| X = = > >Y | $\underline{X} = \underline{Y} = \overline{Y}$ |  |
| X = > = >Y | $\underline{X} = \overline{X} = \overline{Y}$ |  |
| X < = < =Y | $\underline{X} = \overline{X} = \underline{Y}$ |  |

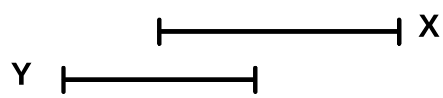
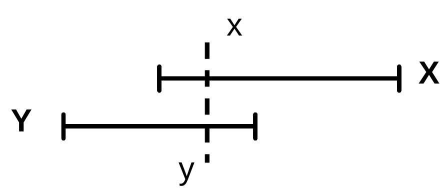
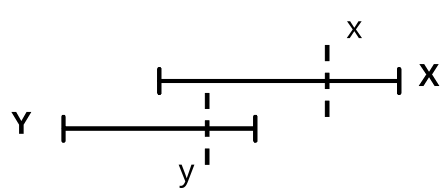
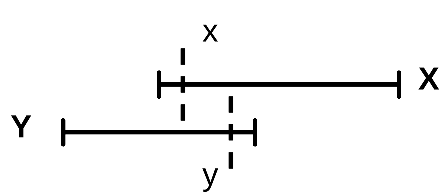
**Table 4. Meaning of interval relations**

<table>
<tr><th>Interval Relation</th><th>Constraint</th><th>Meaning</th></tr>
<tr><td>$X <<<> Y$</td><td>$\underline{X} < \underline{Y} < \overline{X} < \overline{Y}$</td><td></td></tr>
<tr><td colspan="3">Meaning of each component in the relation</td></tr>
<tr><th>Component in the relation</th><th>Scalar comparison</th><th>Meaning of Component</th></tr>
<tr><td>$X <<<> Y$</td><td>$\underline{X} < \overline{Y}$</td><td></td></tr>
<tr><td>$X <<<> Y$</td><td>$\underline{X} < \underline{Y}$</td><td></td></tr>
<tr><td>$X <<<> Y$</td><td>$\overline{X} < \overline{Y}$</td><td></td></tr>
<tr><td>$X <<<> Y$</td><td>$\overline{X} > \underline{Y}$</td><td></td></tr>
</table>

One should note that although these relations determine the relation of the intervals, most of the relations do not impose any constraints on the instances of each interval. Specifically, for any of the relations that involve overlapping intervals, there is nothing that could be said about the relation of their instances. For example, consider $\mathrm{X} <>>> Y$, for the instances of those intervals, $x \in X$ and $y \in Y$, no single relation can be inferred. This concept is shown in Table 5. At first one may think $x > y$, as *x* must be greater than *y* just by the relation of the interval from which they come. But this would be a misinterpretation of the relation. The relation $\mathrm{X} <>>> Y$ means that *x* has the possibility of being greater than *y* has the possibility of being, but will not necessarily be greater than *y*. Nor is *x* known to be more likely to be greater than *y*. In other words, $x > y$, $x < y$, and $x = y$ are all possible, and one does not know which one is more likely. This is a crucial aspect in comparing instances based on their intervals: if the intervals overlap then nothing can be said about the relation of their instances.

Since most interval relations involve overlap, only for the $\mathrm{X} >>>> Y$ and $\mathrm{X} <<<< Y$ interval relations could one determine a relation between the instances in the intervals. In other words, if $X >>>> Y, x \in X, y \in Y$ then $x > y$. These comparisons will be revisited in Chapter 4. With this understanding of interval representation and computation, attention is now turned to another uncertainty representation form.

**Table 5. Relation of Instance of Intervals that Overlap**

| Interval Relation | Constraint | Meaning |
|---|---|---|
| X <>>> $Y$ | $\underline{Y} < \underline{X}$ and $\underline{X} < \overline{Y} < \overline{X}$ |  |
| Let $x \in X$ and $y \in Y$ | | |
| **Possible Instance Relations** | **Meaning of Instance Relations** | |
| $x = y$ where |  | |
| $x > y$ |  | |
| $x < y$ |  | |

### 2.1.4 Probability Theory

Probability distributions have been the most used uncertainty formalism (p. 9-2) in (Nikolaidis, Ghiocel et al. 2005)). Since probability has been so widely used, many of the advancements in engineering involve probabilities. Thus, an understanding of probability is necessary to understand these advancements. Specifically, many design and decision methods are based on probability theory (Luce and Raiffa 1957; Keeney and Raiffa 1993; Triantaphyllou 2000; Fernandez, Seepersad et al. 2001; Stirling 2003), as well as some useful simulation techniques. The design and decision methods using probability are covered in Sections 2.2 and 2.3, respectively, while simulation techniques are investigated later in this section. But, first the basis for these advancements is established with a review of probability theory.

The mathematical probability theory was first developed in Jacob Bernoulli’s *Ars Conjectandi* in 1713 (Bernoulli 1713) and Abraham De Moivre’s Doctrine of Chances in 1718 (de Moivre 1756). Following later in that century, Thomas Bayes presented his theory for assessing subjective probabilities (Bayes 1763). This gave rise to the subjective interpretation of probability, through which the probability is based on a human’s degree of belief. This perspective stands in contrast with the frequentist perspective of probability, in which one could only establish the probability of an event through near infinite data (Neyman 1937; Neyman 1977). These theoretical works provide the rigor that the practical probability distribution is based on.

**Probability Distributions to Represent Uncertainty**

As compared to intervals, probability distributions do not only include what outcomes are possible, but also the probability of each outcome. If a sample space, $S$, denotes all the possible events, $E$, then a probability can be associated with each $E$ that satisfies the following conditions:

(i) $0 \le P(E) \le 1$

(ii) $P(S) = 1$

(iii) For any mutually exclusive event, $E_n$ : $P\left(\bigcup_{n=1}^{N} E_n\right) = \sum_{n=1}^{N} P(E_n)$

where $P(E)$ is the probability of the event *E* (Ross 1997; Devore 2000). This is best explained by an example.

Consider the roll of a six-sided die. In this case, the sample space is $S = \{1,2,3,4,5,6\}$. If the die is fair – all numbers are equally likely to appear – then the probabilities are $P(\{1\}) = P(\{2\}) = P(\{3\}) = P(\{4\}) = P(\{5\}) = P(\{6\}) = 1/6$. Based on (iii), the probability of getting an even number is $P(\{2,4,6\}) = P(\{2\}) + P(\{4\}) + P(\{6\}) = 1/2$.

These definitions, based on a discrete space, help to give the reader a basic understanding of probability; however, in engineering, most of the applications of probability are not on discrete events. Instead, an engineer often is interested in continuous random variables, which are functions defined on a continuous sample space. A continuous random variable has a probability density function $f(x)$, defined for all real $x \in (-\infty, \infty)$ such that for any set of *B* real numbers:

$$P\{X \in B\} = \int_B f(x)dx$$

Thus, the probability that the random variable *X* will be in *B* is obtained by integrating the probability density function over the set *B*. Any number of different probability density functions could be used that fit the definition above; an example is the normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

where $\mu$ is the mean and $\sigma^2$ is the variance. This probability density function gives the classic bell curve. The characteristics of this bell curve are altered by adjusting the mean and variance.

**Probability Distributions to Represent Uncertainty**

The probability distribution could be and has been used to represent both aleatory and epistemic uncertainty. Since aleatory uncertainty is stochastic in nature, it follows some underlying probability, thus a probability distribution accurately reflects this aleatory uncertainty. For this reason, aleatory uncertainty is mainly represented by a probability distribution.

Because aleatory uncertainty is so well represented by a probability distribution, the representation has been applied extensively in engineering, decision theory, and operations research (Luce and Raiffa 1957; Walley 1991; Keeney and Raiffa 1993; Chen, Allen et al. 1996; Chen, Allen et al. 1996; Triantaphyllou 2000; Stirling 2003). An example of this in engineering design is the distribution of material properties. These properties vary randomly based on the unpredictable crystal structure that occurs naturally due to the nature of the manufacturing process. The use of this representation is not limited to materials properties but extends throughout much of engineering; probability distributions are used to represent machining error, variability in environmental conditions, and variables relating to human operation.

Epistemic uncertainty is not stochastic in nature – a probability distribution does not underlie the process. However, engineers often also employ a probability distribution to this uncertainty. Even though this practice is theoretically sound in the subjectivist perspective (Anscombe and Aumann 1963; Schmeidler 1989; Hazelrigg 1996), the designers are adding information that is simply not supported. Using this information in decisions can lead to a less desired result (Aughenbaugh and Paredis 2005).

In this thesis, I do not use probability distributions to represent epistemic uncertainty, instead I use intervals. Although interval contain less information that distributions, I demonstrate in Chapter 4 how these interval can be used to eliminate designs effectively. Even though probability distributions will not be used in this work, some of the methods of computing with probability distribution are drawn from for inspiration. These are reviewed in the next section.

**Computing with Probability Distributions**

Probability distributions are even more difficult to compute with than intervals. Simulations that include distributions often do so by using Monte Carlo methods. These simulations define the inputs as random variables and propagate these random inputs through the system model to output performance in repeated execution of the model. Each time the simulation is executed, a different random number is generated for each input. The performance results from numerous simulation executions allow one to construct an approximate distribution of the system performance output of a system for the uncertain inputs (Buede 2000; Law and Kelton 2000; Fishman 2001). More simulation executions produce a better approximation of the resulting distribution.

Although both Monte Carlo methods and Interval Analysis compute with uncertain information, the two methods are significantly different. Monte Carlo is a practical means of computing with probabilistic inputs that produces an approximate distribution. Monte Carlo does not produce the actual resulting distribution, whereas Interval Analysis is a precise calculation of the bounds on the calculations and results in rigorous bounds.

Monte Carlo simulations do have some advantages over Interval Analysis, thus some concepts are borrowed from Monte Carlo to apply to intervals in this thesis. Specifically, interval parameters in the system performance models are not propagated with interval analysis, instead samples from the interval are used to determine the effect of each uncertain parameter on performance. These effects are used to bound system performance for the most extreme situations described by the intervals. The details of this process are given in Chapter 5.

In many cases, Monte Carlo simulations are performed with the objective of determining the most preferred design alternative. If the simulation for each of these alternatives is generating random numbers to simulate performance then comparisons between the alternatives may be inconclusive because of the introduced randomness, thus one may need more simulations to determine which design is clearly superior.

To address this issue, the modelers have modified the simulations slightly to incorporate Common Random Numbers (CRN), meaning that some of the random numbers used in testing one alternative are used to test the others. Since the same random numbers are used in the simulation of each alternative, the alternatives are compared under the same uncertain conditions. This allows one to determine which alternative is superior with fewer executions of the simulation (Law and Kelton 2000; Fishman 2001).

For example, if one is considering the selection between design alternatives *A* and *B*, one is concerned with difference in the performance: $U(A)-U(B)$. If the performance of both *A* and *B* are distributed, then the results of the Monte Carlo simulation will depend on the variance of performance of each alternative. The variance on performance of the two alternatives is given by the following equation:

$$\operatorname{var}\left(U(A)-U(B)\right)=\operatorname{var}\left(U(A)\right)+\operatorname{var}\left(U(B)\right)-2\operatorname{cov}\left(U(A),U(B)\right)$$

where $\operatorname{var}(U(A))$ is the variance on the performance of *A*, $\operatorname{var}(U(B))$ is the variance on the performance of *B*, and $\operatorname{cov}(U(A),U(B))$ is the covariance between the performance of *A* and *B*. With typical Monte Carlo simulation, where the simulations are performed without common random numbers, there is no correlation between the alternative performance and the variance of the difference between the two alternatives' performance is:

$$\operatorname{var}(U(A)-U(B))=\operatorname{var}(U(A))+\operatorname{var}(U(B))$$

Thus, typically Monte Carlo does not include the covariance (Law and Kelton 2000; Fishman 2001), which has no impact when there is no correlation between the performance of the two alternatives, but there is in many cases. In these situations, one can take the covariance into account by using Common Random Numbers (CRN). CRN involves coordinating the simulations such that the same random numbers are used in all the alternatives considered, thus the covariance is considered in the simulations. Thus, the variance in the performance difference is smaller, allowing one more easily to determine which alternative is superior.

Because the concept behind CRN is valuable when comparing alternatives under uncertainty, it is extended in this thesis. The correlation between uncertainties in different alternatives is considered in the more general case, and termed Common Uncertainty; this work is presented in Chapter 4.

### 2.1.5 Summary of Uncertainty Review

In this section, I have distinguished between aleatory uncertainty and epistemic uncertainty. Engineers have been quick to recognize aleatory uncertainty, but epistemic uncertainty and the complications it causes have not been recognized distinctly. This is problematic because epistemic uncertainty is common in engineering – one needs to consider this uncertainty when designing and making decisions.

To consider this uncertainty, one first must represent the uncertainty. While probability distributions have been used to represent epistemic uncertainty, this representation could lead one to make inappropriate decisions, as pointed out above. Instead of taking this approach, I represent epistemic uncertainty with intervals. While this is a gross oversimplification of the problem, valuable insight can be gained by doing so. The knowledge gained by eliminating with this information may be applied to other uncertainty representations. Specifically, this knowledge can hopefully be applied to imprecise probabilities, as this representation leads to intervals on the expected utility (Walley 1991). Thus, assuming intervals for epistemic uncertainty is useful for advancing the engineering design knowledge.

For obtaining these intervals, the engineers involved should consider the value of the representation. An interval that is too wide can result in an inability to make decision, whereas an interval that is too narrow can result in the wrong decision. These outcomes need to be balanced based on their value (or cost) to the company.

The expressiveness of the interval representation is least among all the representations. This could be a drawback if more information is known about the uncertainty – for example, if one knew the probability distribution. Valuable information is lost, making an engineer's job of making decisions significantly more difficult. However, with the epistemic uncertainty in design, one may not have additional information to express, thus a more expressive representation may not be any more useful in design.

I approach the problem with a conservative representation of epistemic uncertainty using intervals. I demonstrate that these intervals can be used to progress significantly in the design process without the risk of eliminating designs that could potentially be the most preferred. The success of this conservative approach is presented in Chapter 4. For now, the reader's attention is turned to a few different approaches to design.

## *2.2 Approaches to Design*

In the previous section, characteristics and representations of uncertainty were reviewed. This review points out some of the issues that are caused by uncertainty and must be accounted for in design approach under uncertainty, as is the larger goal of this research. In this section, this knowledge about uncertainty is used to review approaches to the design process for their usefulness in designing under uncertainty. While developing a set-based approach to design under uncertainty is the larger research objective, these design processes also are reviewed to determine a set-based design process to use as the context to view elimination in this thesis.

Based on the review of uncertainty and design processes, I believe that in order for a design process to handle the challenges of uncertainty successfully it should address some basic criteria. Specifically, I see a design process that handles uncertainty meeting the following:

1. **A design process must provide a mental structure for the designers.** Every process carried out by humans needs to make the process structure explicit – humans have to understand why they are doing what they are doing, and they need to get in the right mindset to do so. This is recognized by Pahl and Beitz: "Systematic procedures help to render designing comprehensible and also enable the subject to be taught." Pahl and Beitz then continue, "Systematic procedures merely try to steer the efforts of designers from unconscious to the conscious and more purposeful paths (p. 11 in (Pahl and Beitz 1996))." Thus, a design approach should be systematic, spelling out the actions for the

designers to take explicitly, eliminating any potentially wasteful actions. The designers should know their actions and how those actions fit into the overall process.

2. **A design process must adequately account for uncertainty.** Structure in the design process helps a designer, but that structure needs to incorporate and adequately represent uncertainty for the designer to make the design decisions. Peter Walley sees this point: “Reasoning begins with the recognition of ignorance and uncertainty.” (p. 1 in (Walley 1991)) Thus, a good design method should recognize and adequately handle the uncertainty in design.
3. **The method should require the designers to verify performance.** Designers cannot accept a design without knowing whether that design works; the design has to be verified in the process. Forsberg and Mooz recognize this need and specify that the design verification should be an integral part of the process, supported by specifications at each level of abstraction in the design (Forsberg and Mooz 1991; Forsberg and Mooz 1996). To minimize the resources spent in verification, the verification should be planned and prepared for throughout the design process (Buede 2000).
4. **Lastly, a good design method should show internal consistency.** The goal of the design process is to find the most preferred design, and each step in the process should move closer to this goal; the process must converge on the most preferred design. The process needs to show this internal consistency to be desirable to use. Pahl and Beitz point out this need to have a process that can be verified (Pahl and Beitz 1996).

I see these criteria as necessary to handle the interval-based uncertainty faced in design while closing in on the most preferred solution. These criteria are used to review the approaches to design, beginning with the Pahl and Beitz Systemic Design Methodology.

### 2.2.1 Pahl and Beitz

Pahl and Beitz saw the need for a systematic approach to design: “In the design process, the required design activities have to be structured in a purposeful way, that is in a clear sequence of main phases and individual working steps, so that the flow of work can be planned and controlled (p. 11 in (Pahl and Beitz 1996)).” Pahl and Beitz took the first steps in formalizing the design process by studying German designers and extracting their methods into a systematic approach to engineering design. Their approach is meant to act as scaffolding to support the search for solutions and evaluations throughout the design process, rather than a definitive, product-specific method. A crucial component in this general approach is the general phases and tasks that Pahl and Beitz recognize; these phases and tasks are shown in Figure 2.1. The main phases of this process are as follows:

1. Project Planning and Clarification of Task
2. Conceptual Design
3. Embodiment Design
4. Detail Design

For each of these phases there are specific tasks and resulting milestones that Pahl and Beitz include in their design process. Pahl and Beitz give guidance, indicating how each of these tasks should be carried out and the aspects of design that should be taken into account in each. The result is a systematic design process with significant flexibility.

Although the Pahl and Beitz approach is systematic at the high-level, providing specific phases and tasks to be carried out, the approach does not provide a systematic method for handling the epistemic uncertainty inherent in design. In fact, Pahl and Beitz fail to recognize uncertainty other than a few brief remarks about how one should consider risk in design.

In addition, the Pahl and Beitz approach does not necessarily progress toward the most preferred design, instead the process suggests iteration, but does not instruct how this iteration should be used to search the design space, or if the iteration will ever converge. Designers must determine how to search the design space. Iterating in this manner could result in wasted effort with no hope of convergence.

Lastly, P&B does not provide an explicit method for verifying that the design artifact meets the requirements. Rather, verifying the design in a formal manner is more of an afterthought in the P&B methodology. An internally consistent model for design should provide a framework in which the designed system is validated down to the operation of the subsystems. Such an internally consistent model for design is already in practice; this model is reviewed in the next section.

### 2.2.2 System Engineering V-model

The Pahl and Beitz design process was created by mechanical engineers for mechanical designers and is therefore best suited for the design of mechanisms or mechanical systems in general. The approach in the Vee Model of Design originates in systems engineering. Systems engineers typically design systems of larger scope, including multiple sub-systems and involving multiple disciplines. The nature of their field has forced systems engineers to take a more top-down approach and maintain an underlying structure that demands internal consistency and design verification.

These demands are incorporated into the Vee Model of Systems Engineering, introduced by Forsberg and Moos (Forsberg and Mooz 1991; Forsberg and Mooz 1996) and diagrammed in Figure 6. The Vee Model trisects the engineering process of a system into the main components represented by the left side, bottom, and right side of the Vee; these represent the Decomposition and Definition, Discipline Design, and Integration and Verification phase, respectively.

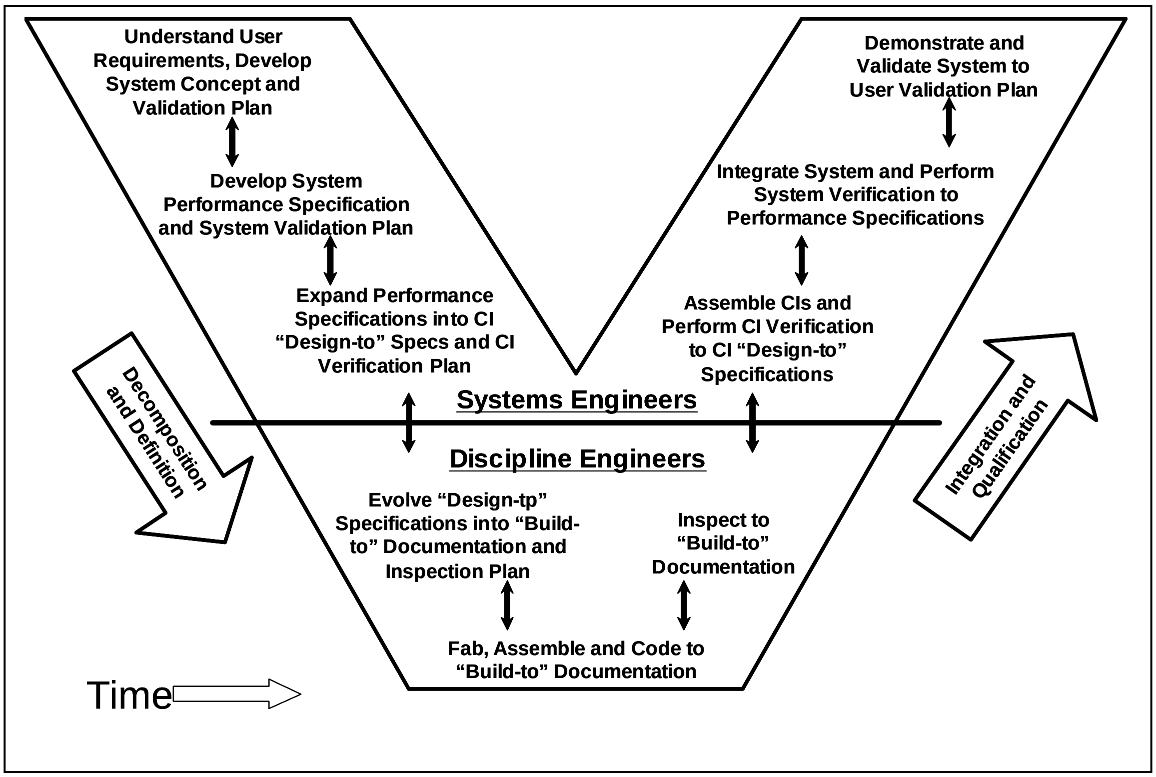



*Figure 6: Vee Model of Systems Engineering (Forsberg and Mooz 1991; Forsberg and Mooz 1996)*

In the Decompositions Phase, system specifications are decomposed progressively from the highest system-level specifications until further decomposition is not valuable. These final specifications are on the sub-system requiring discipline design, and are known as Configuration Identifications (CI's). Numerous CI's are the result of the Decomposition Phase and the input to the discipline design phase. The discipline design of the CI is handled by engineers from that particular discipline. The CI is embodied and detailed by the engineers just as one would do in the Pahl and Beitz design process, resulting in production, or “build to” documentation for each sub-system.

In the Integration and Qualification Phase, the production documentation from each sub-system is used to determine performance, by realizing or simulating the sub-system, and this performance is compared to the CI’s. By this process the designed sub-systems are verified. In this phase, verification continues up the hierarchy by composing the system at each level, determining the performance, and then comparing the performance of that system to the specifications from decomposition. When this phase reaches the highest system-level, the engineering process is complete and the resulting design is verified and ready for commission.

By decomposing, designing, and validating the engineered system in such a manner, the Vee Model of systems engineering certainly meets the requirement of verifying the design throughout the process. Unfortunately, this top down approach misses the bottom-level problem: how does one determine what requirements lead to the most preferred design? There is no method for handling this implementation-level problem or for determining, in the discipline design, how one determines which designs lead to the most preferred system. In addition, the Vee Model fails to adequately incorporate uncertainty; one surely could include uncertainty in their specification but there is no guidance as to how to do so. In contrast, the next design approach reviewed, Set-Based Concurrent Engineering, gives principles for handling uncertainty and narrowing design sets.

### 2.2.3 Set-Based Concurrent Engineering (SBCE)

The previously reviewed design approaches have not specifically, or adequately, incorporated uncertainty. Uncertainty in the design process lends itself to a set-based approach to engineering, as implemented at Toyota in the form of SBCE. Although an informal approach, SBCE gives general principles to designing under uncertainty and has proven highly successful at Toyota.

The Toyota Motor Company experienced success and growth in market share over the past twenty years in an increasingly competitive automotive market. This success has warranted the attention of academia and industry for the widely acclaimed 'Taguchi' method of quality control (Taguchi 1987). The Taguchi method has not only been accepted by academics but has been widely built on in developing new quality improvement methods, speaking to the effectiveness of these quality methods. But Toyota's competitive advantage not only came from Taguchi's approach to quality but also from their set-based approach to engineering.

Set-Based Concurrent Engineering (SBCE) has been as much a part of Toyota's success as the Taguchi method; just as with Taguchi methods academia and industry have taken notice of this Set-Based Approach (Chang 1994; Ward, Liker et al. 1995; Liker, Sobek II et al. 1996; Sobek and Ward 1996; Finch 1997; Finch 1997; Parunak, Ward et al. 1997; Simpson, Rosen et al. 1997; Wu 1999; de Weck 2002; Costa and Sobek II 2003; Ford and Sobek II 2004). Now Set-Based Concurrent Engineering (SBCE) has been praised as playing an integral part in Toyota's success. Toyota's advantage gained through SBCE are pointed out by Ward (Ward, Liker et al. 1995): "[SBCE], in brief, delaying decisions, communicating 'ambiguously', and pursuing excessive number of prototypes, enable Toyota to design better cars faster and cheaper." At first thought, it seems that these practices of delaying decisions, communicating ambiguously, and producing an excessive number of prototypes are inefficient. But these practices are actually extremely efficient when one considers the deeper ideas behind SBCE.

To delve deeper into SBCE and how it works, attention is turned to the work of Ken Sobek, who observed the process and recorded what he observed (Sobek and Ward 1996; Sobek II 2004). From his observations, Sobek determined 12 principles behind the SBCE process at Toyota. These principles are as follows (Sobek and Ward 1996):

1. **Define feasible regions.** Specialized engineers look at the problem from their differing perspectives, and from their perspectives, they define what is critical for the design to have and what is desirable. This defines the broad feasible regions of the design with which they begin the design process.
2. **Communicate sets of possibilities.** Throughout the design process the design teams communicate in terms of feasible regions, not in terms of point solutions or best alternatives. The set of possibilities can be represented in many ways depending on the state of the design, including requirements, desires, or specifications.
3. **Look for intersections.** After examining the sets defined by the other specialized teams, the engineering teams reduce the feasible set by looking for commonality in their sets. The specialized engineers involved in the project have significant uncertainty with respect to the other perspectives; therefore they do not reduce their set without conforming to the feasible sets specified by the other specialized engineers. In this way specialized engineers become more certain of the other aspects of the design; uncertainty about the other aspects of the design are reduced.
4. **Explore trade-offs by designing multiple alternatives.** In SBCE, engineers investigate all possibilities in the feasible set by designing and testing these alternatives. In doing so, they gain a better understanding of these possibilities, as well as the advantages and disadvantages they offer. Thus, they know the trade-offs and become more certain of the

decision they make. In addition, the experience the engineers gain from exploring is useful in their future projects.

5. **Impose minimum constraint.** Engineers are not to impose more constraints on the feasible sets than are necessary. This allows designers to make adjustments in future design decisions, as they become more certain of the design. For example, this principle actually includes the finalized design, in which tolerances are not always given to the manufacturing engineers. Instead, nominal dimensions are provided with the idea: "make the parts function as they should; use the nominal dimensions as a guide." The ambiguity allows the manufacturing engineers to make adjustments as necessary to the manufacturing process; in other words, the designers are uncertain about the exact best dimensions so they do not impose those as constraints, instead they give general guidelines along with the overall objective.
6. **Narrow sets smoothly, balancing the need to learn with the need to design.** In SBCE, the engineers are encouraged to explore and learn about their possible solutions before narrowing the set. In addition, engineers are encouraged to spend time investigating and to avoid deciding on a single solution too quickly. However, the engineers must narrow the set in a reasonable time, thus there is a balance that needs to be maintained between the need to learn and the need to decide. The balance is typically maintained through the supervision of the chief engineer who will determine if further investigation and learning is necessary. Recognizing this trade-off between learning and deciding is an important point of SBCE.
7. **Pursue high-risk and conservative options in parallel.** Both high-risk and conservative alternatives are developed as feasible solutions. This allows for a possible

breakthrough of a high-risk design, while not taking the risk involved in relying on that high-risk design as the only solution.

8. **Establish feasibility before commitment.** Design teams will not commit to any single alternative until they are sure that alternative works. When a solution is finalized the designers know, from extensive prototype testing, that the solution works and is acceptable, otherwise they do not settle on that alternative.
9. **Stay within sets once committed.** This principle ties in closely with the last, as designers can be confident that they have a working solution there is no reason to backtrack in the design process. Since they have investigated all of the alternatives, they are confident they have the most preferred alternative.
10. **Control by managing uncertainty at process gates.** Toyota requires each engineering team to report on the feasibility and size of the set they are considering. The uncertainty in both the set and the performance must conform to the project time frame. This principle is closely connected with principle 6. This principle again highlights the trade-off between learning and deciding.
11. **Seek solutions robust to physical, market, and design variation.** Taguchi methods and other robust methods apply to this aspect of SBCE. Through these methods designs are as robust as possible to all uncertainty.

These principles appear to address the problems that uncertainty causes in design. Because of the large uncertainty in design, the designers may not be able to distinguish which alternative is the most preferred design. In this situation, the engineers do not converge on a single alternative, instead they learn about the tradeoffs of each alternative in the feasible set. As the engineers learn more, they reduce the uncertainty and eliminate alternatives. This appears to be the crux of SBCE: engineers should only make decisions supported by the information that they possess and learn more about decisions for which they don't have the sufficient information.

This may seem like an obvious principle but it is very often ignored. Instead designers typically decide on what they believe to be the best solution, while ignoring other promising solutions. Selecting a single best alternative would work if the designers are certain that alternative is the best solution, but such certainty is rare given the uncertain information in design. The alternative that the engineers select as the 'best' typically is less than the best. Since the engineers have not investigated the other feasible alternatives, they would not know that their 'best' alternative is less than the best, thus they continue the design process without knowing they are working with an inferior alternative. If the engineers discover that their choice was not the best decision, they could continue with their current design and accept the loss in performance, or they can start over again without any knowledge as to what other options could work. Both of these possible scenarios are wasteful, and as pointed out in Chapter 1, the entire process rarely leads to the most preferred design.

Designing in this iterative fashion is inefficient and ineffective as pointed out by Sobek (Sobek and Ward 1996): "When a function proposes only its best idea it does not give other functions a clear idea of the possibilities, so the iterative process is likely to involve much waste." The wastefulness of iterating becomes apparent to experienced designer who thought after a design process: 'why didn't we try that in the first place'. With SBCE they do try that in the first place, as those possibilities are considered in the feasible set; that is the advantage of SBCE.

The advantage provided by a set-based approach to design has been recognized by other researchers, both aware and disconnected from the Toyota approach (Chang 1994; Ward, Liker et al. 1995; Liker, Sobek II et al. 1996; Finch 1997; Finch 1997; Parunak, Ward et al. 1997; Wu 1999; de Weck 2002; Costa and Sobek II 2003; Ford and Sobek II 2004). All have pointed out the advantages of considering sets, rather than points, and leaving design freedom in the problem. Leaving the design freedom in the design problem has also been a battle cry of other researches, as this allows more flexibility for Mass Customization, as pointed out by Simpson (Simpson, Rosen et al. 1997).

Although SBCE has been applied successfully at Toyota, there is still no formal specification of the process. First, the engineers in SBCE are required to verify that an alternative works before eliminating all other alternatives, but how the engineers verify an alternative is not clear, nor formal. Second, there is nothing in the current state of SBCE that says how and when the set of possibilities should be reduced by the engineers. This shortcoming is recognized by (Sobek and Ward 1996): "Deciding when to decide becomes a central task of the chief engineer. Do we really know enough to make a decision? Or should we eliminate some more, and develop the rest a little further and decide next week?" This problem only has been addressed in a brief, informal manner by researchers, one which makes many simplifying assumptions (Wu 1999), and another which uses predicate logic to eliminated designs based only on constraints (Finch 1997; Finch 1997). Lastly, the process, as presented by Sobek, has not been organized into phases that provide the designers with an understanding of the purpose of their actions. Despite this lack of formalism, the design process provides good motivation for building a successful design process that is geared for handling uncertainty.

### 2.2.4 Robust Concept Exploration Method (RCEM)

A method for developing solutions robust to design uncertainty was developed in the form of the Robust Concept Exploration Method (RCEM) (Chen, Allen et al. 1996; Chen, Allen et al. 1996). The method is based on an algorithm that integrates multiple engineering tools to develop robust solutions to the formulated problem. These solutions are robust to the aleatory uncertainty recognized as originating from controls factors and noise factors and modeled with probability distributions. The method creates these robust solutions by weighing the variability of the design performance in the objective function, thus designs with lower variation in performance (more robust) are preferred.

This approach to generating solutions makes sense for conditions of aleatory uncertainty; however in this approach, future design decisions are considered a noise factor and modeled with a probability distribution. In doing so, this approach favors design alternatives that are robust to variations in future design variables. The justification for this approach is the following: Since the designer lacks knowledge about the outcomes of future design decisions, the designer should make the current decision in a way that yields a satisficing solution no matter what the outcome is of the future design decisions. This is a reasonable approach if the price one pays for robustness is relatively small; that is, if the value of the most preferred solution is reasonably close to the value of the robust, satisficing solution. Unfortunately, if significant decisions still remain in the design process, this difference in value could be quite large because decisions that significantly affect performance still need to be made. In addition, the robust design methods do not provide any indication of how large that price is, nor could one determine that without knowing the most preferred design.

RCEM not only provides a questionable approach to handling future design decisions, this method is not a complete design method. First, the method is successful at generating alternatives, but lacks some means to verify these alternatives; that is left for the designers to do on their own. Second, because the method is point-based the search for a robust solution could likely leave out solution space that contains that solution. Lastly, the method uses a complicated structure that is difficult for inexperienced designers to understand. Thus, it is not easily learned or applied by engineers.

Based on the methods I have reviewed, there is still a need for a formal design process that is geared for handling uncertainty. Since the design methods I have reviewed do not provide the formal structure, I review the Branch and Bound Algorithm as a basis for that formal structure.

## *2.3 Branch and Bound Optimization Algorithm*

Since the design approaches available do not meet the need for formal treatment of the uncertainty in design, attention is turned to a close relative to design processes for an answer: optimization algorithms. Optimization algorithm can be closely likened to the process of design, as both involve a search for the best possible solution (p. 1 in (Belegundu and Chandrupatla 1999), p. 1 in (Pahl and Beitz 1996)); this similarity has led to the use of optimization in the design process. This application of optimization algorithms in design is not the focus of this section. Rather, the Branch and Bound Global Optimization Method, founded in Interval Analysis, is reviewed for its applicability to the set-based design process under conditions of uncertainty, which serves as the context for the elimination method developed in this thesis.

Although this review is focused on the Branch and Bound Algorithms, there are some other set-based optimization approaches that could incorporate uncertainty and could be considered for set-based design. One such set-based optimization approach is the Genetic Algorithm. This algorithm considers a set of finite alternatives that are evaluated based on their fitness. The inferior elements in the set are eliminated before the elements of the sets breed to create a new set of elements. The process then continues as those elements are evaluated for elimination (Goldberg 1989; Davis 1991; Belegundu and Chandrupatla 1999). This approach has three main problems. First, the process must have a finite set of alternatives, which does not allow the set to contain all of the possible most preferred designs. Secondly, the process eliminates alternatives when creating the next generation that could be the most preferred. Finally, because the process does not necessarily contain the most preferred alternative in the set, the process is not guaranteed to converge on the most preferred design. While the modified versions of the algorithm could converge on the most preferred alternative, this could require a large number of computations.

Another optimization algorithm that works on elimination is the Tabu Search. This algorithm searches point by point as more traditional approaches to optimization; however, the algorithm keeps track of inferior points so that those points are not evaluated again needlessly (Glover 1986). While this solution method could be adapted for use with sets and under uncertainty, it does not provide a structure that is conducive to including these elements. Additionally, the algorithm may not converge on the most preferred solution. Since both Genetic Algorithms and Tabu Searches have deficiencies for applying them to set-based design, this section focuses on the Branch and Bound (B&B), which I believe provides the most-promising paradigm for a formal set-based approach to design. The advantages of using this paradigm for set-based design are pointed out at the end of this section and in Section 3.1. Before stepping into these advantages, the details of the algorithm are explained first.

Branch and Bound (B&B) represents an entire class of algorithms as pointed out by Clausen (Clausen 1997): "B&B is, however, an algorithm paradigm, which has to be filled out for each specific problem type, and numerous choices for each of the components exist." This class of algorithms contains common steps that vary slightly in the implementations; these general steps in the method are explained in Figure 7 and diagrammed out in Figure 8. The region being searched is divided, or branched, to create multiple sub-regions. The sub regions are then characterized by bounds on performance so that then, based on those bounds, the regions can be eliminated from the search. The process continues by choosing which of the remaining sub-region(s) to focus the search on; these selected sub-regions are branched, beginning the next iteration. The process continues until the solution is obtained within an acceptable limit of either performance or sub-region size (Nemhauser, Rinnooy Kan et al. 1989; Clausen 1997; Montemanni, Gambardella et al. 2004). To investigate this process deeper, attention is turned to the individual steps in the process.

Starting with a solution space, S, and searching for the maximum performance, as determined by the objective function value, U, the general steps of the algorithm are as follows:

**Step 1** Branch the given set. The set should be divided into subsets according to the specifics of the algorithm.

**Step 2** Bound the performance of the subset(s). Each sub-region should be characterized by conservative bounds on that sub-region's performance: $U = \left[\underline{U}, \bar{U}\right]$. The order in which these are evaluated is determined by the algorithm's search strategy.

**Step 3** Eliminate subsets that cannot contain the best solution; this is determined based on the subset performance bounds.

**Step 4** Select the subset(s) to branch and evaluated next. This is determined based on the search strategy and the subset performance bounds.

**Repeat** steps 1-4 until region is reduced sufficiently or until performance bounds on the remaining regions are sufficiently tight.

*Figure 7: Steps in the B&B Algorithm*

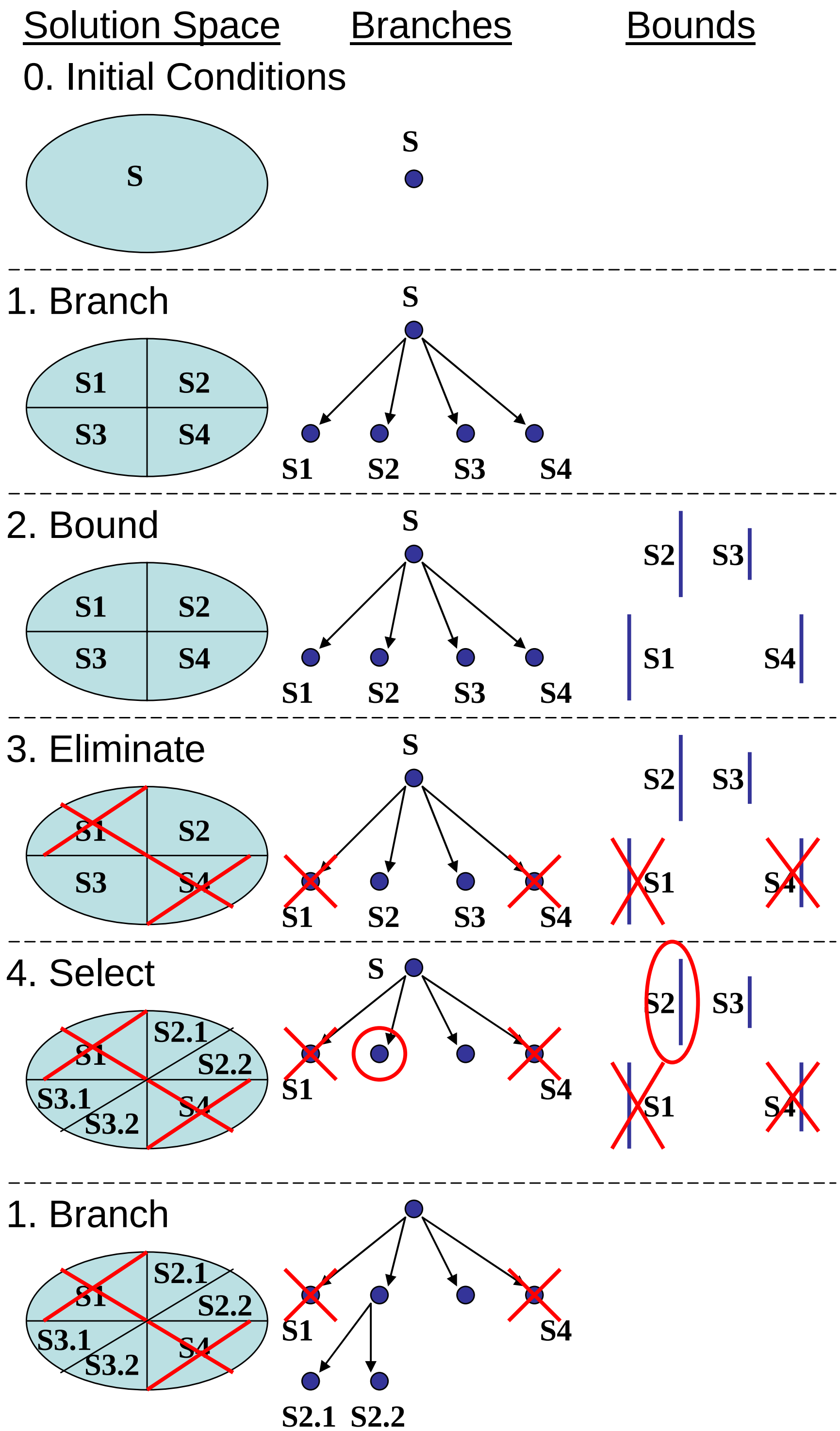



*Figure 8: Steps in the B&B Algorithm with the impact of each step on the solution space, branches, and bounds*

### Branching Step

In the branching step, a chosen set of the solution space is divided into subsets. The set is chosen based on the search strategy that is covered in the 'Strategy for Node Selection' subsection. The branching can produce two or more subsets, called dichotomic and polytomic branching, respectively; these are presented in Figure 9.

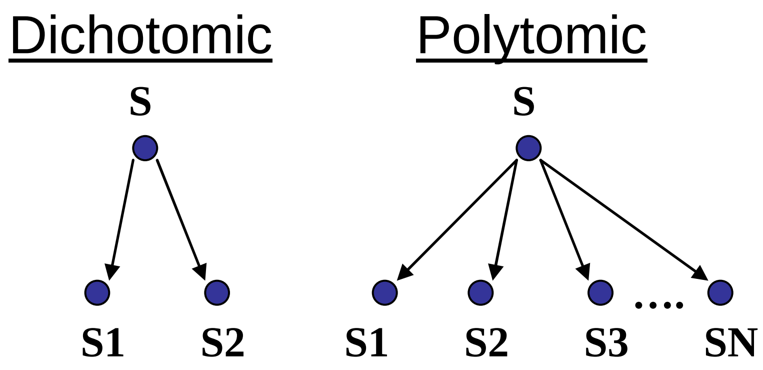



*Figure 9: Dichotomic branching results in two subset, whereas polytomic branching results in more than two subsets*

Whether one uses dichotomic or polytomic branching, each subset created in branching must be smaller than the original set for the theoretical convergence of the algorithm (Clausen 1997; Boyd, Ghosh et al. 2003). This constraint is not difficult to maintain; however, more thought should be put into the practical aspects of branching to ensure practical convergence – convergence with reasonable time and resources. A good branching step can improve the efficiency of the algorithm and may be necessary for practical convergence.

In the algorithm, the branching step serves two main purposes that are interrelated. First, since the subsets are a smaller set of solutions, the solution is specified in more detail in the subsets. This moves closer to the obtaining a completely specified solution and allows for a tighter bounds on the subset, which is the second purpose. These bounds are used in the algorithm to eliminate subsets and to perform the search strategy of the subsets. Tighter bounds allow more dominated subsets to be eliminated and the search strategy to be more effective. Thus in branching, a more efficient step results in subsets that can be tightly characterized.

Subsets that can be bounded tighter are typically smaller, thus in creating subsets with tighter bounds, one may wind up creating an excessive subset. This could be inefficient as well. The tighter bounds offered by smaller subsets need to be balanced with characterizing the numerous, smaller subsets. The result is a need for subsets that can be easily characterized, but are not too numerous that the computation is taxing. In addition, the subsets are typically disjoint; this way the same feasible solution does not appear and is not characterized in different subsets – a wasteful activity.

These objectives for the branching step are based on the results of the bounding step. This suggests that bounding is critical to the success of the algorithm.

## Bounding Step

The bounding step characterizes, or bounds, the performance of a subset such that $f(X) \subseteq g(X)$, where $X$ is the subset, $g\ (X)$ is the characterization, and $f\ (X)$ is the true performance possible in the subset. While one wants to find these conservative bounds, one also does not want bounds that are more conservative than necessary. This creates an optimization problem in itself: the more time and computation spent on bounding, the more accurate the bounds, while the quicker the bounds are computed, the faster the algorithm. This is the tradeoff faced when formulating a bounding function (Clausen 1997).

Bounding step typically works in one of two ways (Nemhauser, Rinnooy Kan et al. 1989; Clausen 1997). First, the step relaxes the constraints on the problem and solves for the extremes. With relaxed constraints, the extremes are found faster and are more extreme than the true values. Second, the step evaluates the extremes for a function that is easier to evaluate and more conservative than the objective function.

These methods are effective at computing conservative bounds on the objective function over the region of the subset; however, it has been recognized that sharp bounds can be computed if the function is monotonic (Hansen and Walster 2004). A monotonic function has a first derivative that does not change sign, thus the bounds of the function are at the boundary of the region being searched. This concept is shown by comparing monotonic functions against a non-monotonic function in Figure 10. The functions on the left of this figure are monotonic, thus the bounds occur at the boundary of the region. However, the function in the figure on the right is non-monotonic, therefore the bounds do not necessarily occur at the boundary. In this case neither bound is at the boundary of the variable range. Monotonic functions occur often in design, especially with respect to uncertain variables, thus this knowledge about monotonic functions is used in this work. This speeds up the bounding process and results in sharp bounds. More advanced methods for computing the bounds on a subset exist (Moore 1966; Moore 1979; Alefeld and Herberger 1983; Broadwater, Shaalan et al. 1994; Kearfott and Kreinovich 1996; Yaman, Karasan et al. 2001; Hansen and Walster 2004), but these will not be discussed here, as they are not applied in this thesis.

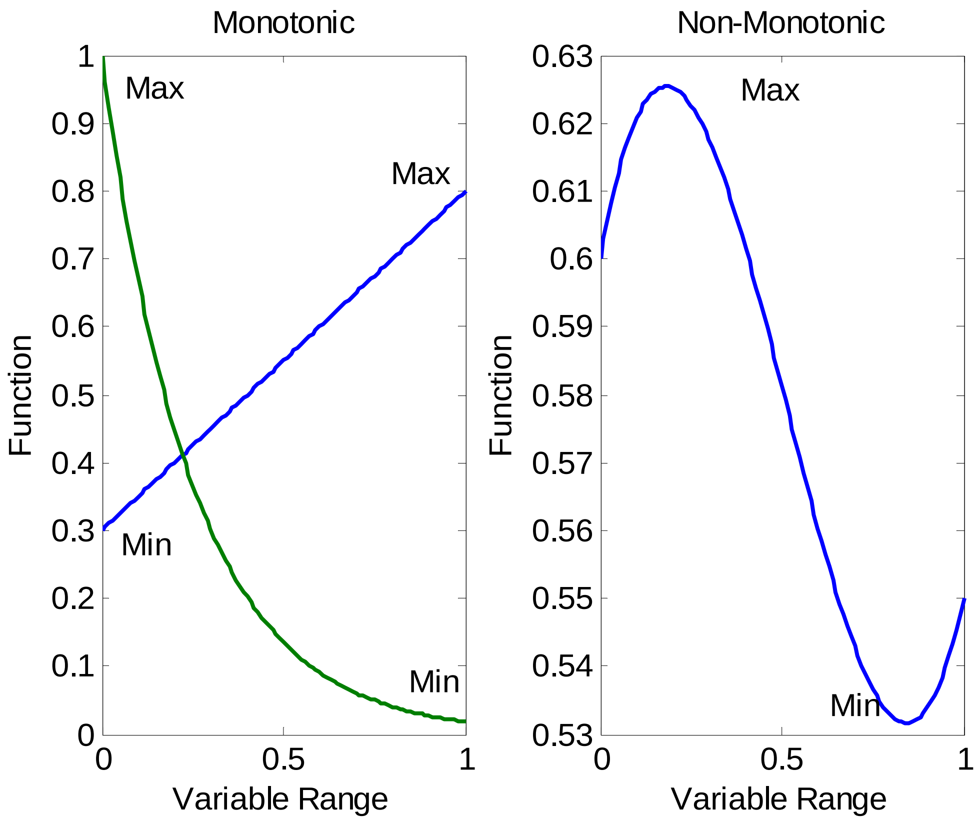



*Figure 10: Steps in the B&B Algorithm and the B&B Design Process are very similar*

These bounds are used by the algorithm to determine which subsets should be branched further and which should be discarded. Because bounding influences the course of action in future steps, the step is extremely important in the algorithm's performance, as recognized by Clausen: "The bounding function is the key component of any B&B algorithm in the sense that a low quality bounding function cannot be compensated for through good choices of branching and selection strategies." (Clausen 1997)

## Eliminating

The bounds on a subset specify the possible performance within that subset. If the bounds on a subset indicate that the solutions in that subset cannot perform as well as at least one other subset then that subset can no longer contain the optimal solution. There is no reason to consider that subset; the subset is eliminated, also known as pruned or discarded. This elimination step increases the available memory and eliminates the subsets no longer worth considering.

Consider the subsets and their bounds given in Figure 11. Based on these bounds, subset 1 and 4 cannot perform as well as either subset 2 or 3; therefore subsets 1 and 4 are eliminated. In most B&B algorithms, a subset is bounded and checked for elimination as soon as that subset is available; such an algorithm is called an *eager* B&B algorithm. Conversely, a *lazy* B&B algorithm creates nodes without bounding or evaluating those nodes for elimination. This lazy approach is employed in some depth-first searches.

In many optimization problems, one does not know or cannot computationally determine the objective function value for the best solution until the search is over. Therefore, the algorithm only can be sure that it has reached the best solution when all other possible solutions are eliminated. Elimination makes convergence possible in these algorithms (Lawler and Wood 1966; Clausen 1997). This would also be true for design, as we shall see.

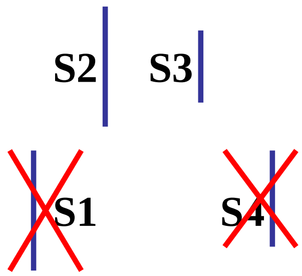



*Figure 11: Based on the bounds, subsets 1 and 4 cannot perform as well as subsets 2 and 3, therefore subsets 1 and 4 can be eliminated.*

**Strategy for the Selection of the Node**

To continue toward the optimal solution, the branching and bounding are repeated on remaining nodes, but the order in which they are repeated significantly effects the efficiency of the algorithm. If that algorithm first investigates subsets that likely do not contain the optimal solution then resources are wasted in doing so, likewise if all the subsets are investigated then this spreads the resources thin. Researchers have recognized the importance of these strategies to the efficiency of the algorithm and have investigated the merits of different search strategies (Clausen and Traff 1991; Laursen 1993; Clausen 1997; Kearfott 1997; Clausen and Perregaard 1999).

There are three main classes of search strategies to address this problem. First, there is the Breadth-First Search (BFS), in which all nodes at one level of the search tree are processed before any node at a higher level. This is shown in Figure 12, which shows the order of evaluation and the resulting evaluation. This searches the range of possibilities, but in larger search trees, where the number of nodes at each level grows exponentially, this search is not efficient and may not even be feasible.

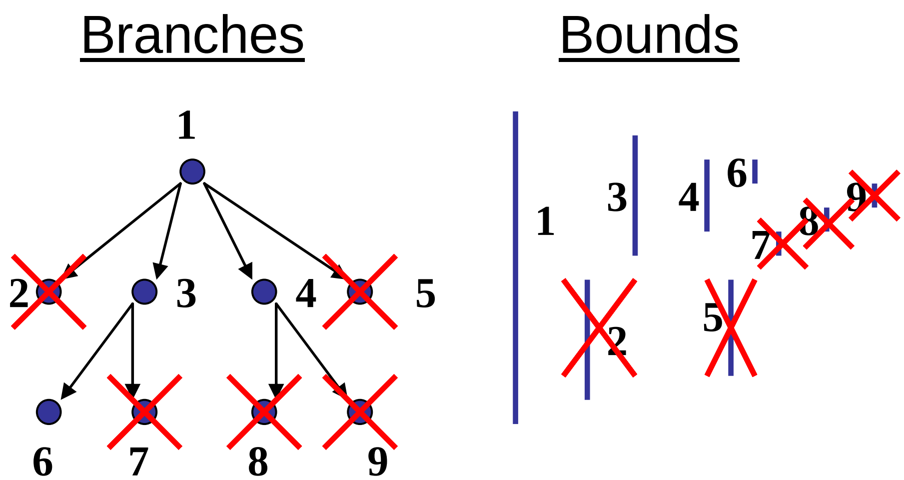



*Figure 12: The Breadth-First Search (BFS) evaluates all the nodes at one level before moving to the next.*

An alternative to this is the Depth-First Search (DFS), in which the algorithm ignores all but one subset until the algorithm has reached the deepest level possible. An example of this search is shown in Figure 13. This example shows how the DFS can be very inefficient, searching too many subsets in depth before finding the best solution, if the wrong subset is shown to search in depth. Poor implementations of the search can wind up searching the solutions exhaustively, defeating the purpose of the intelligence of the B&B algorithm. However, this search can converge quickly on the solution if the correct subsets are chosen at each step of the search with little memory requirements. This is avoided by combining DFS with a selection strategy, which uses improved criterion to determine which subset is the best to investigate. This type of DFS has seen success in execution (Clausen and Traff 1991; Clausen 1997; Clausen and Perregaard 1999).

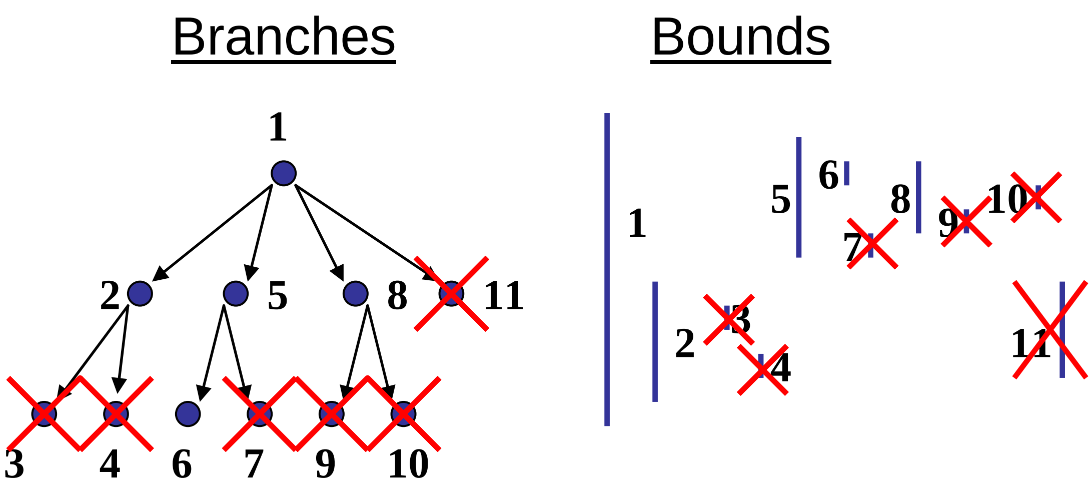



*Figure 13: The Depth-First Search (DFS) focuses on one subset at a time, evaluating that subset to the deepest level possible before moving to the next subset.*

The final class of searches is the Best-First Search (BeFS), in which the algorithm searches the subset with the highest upper bound, no matter at which level of the tree that subset is located. This type of search is diagrammed in Figure 14. The idea with this search strategy is that no superfluous bound calculations take place after the optimal solution has been found. Theoretically, this search would perform the best, but in some tests it has actually shown to perform worse in all situations when compared to a well-formulated DPS (Clausen and Traff 1991; Clausen 1997; Clausen and Perregaard 1999). In addition, the BeFS can run into problems if the bounds are too wide: the search will turn into a BPS and be just as inefficient. This once again shows the importance of good bounding in B&B.

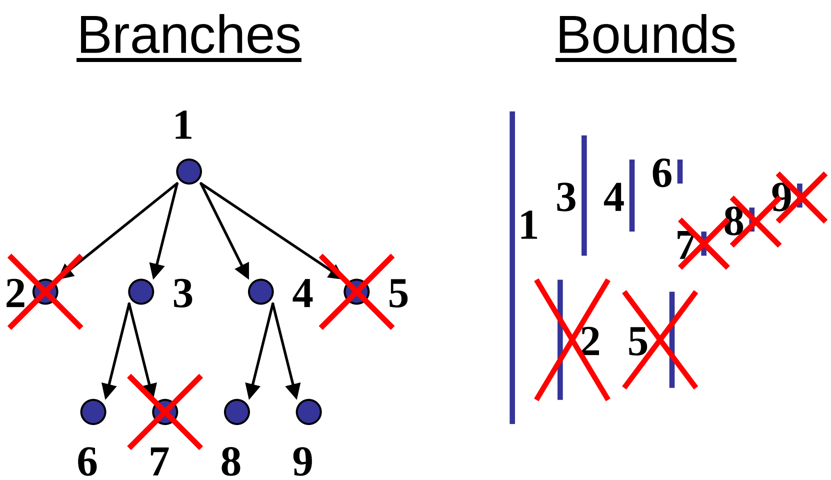



*Figure 14: The Best-First Search (BeFS) branches on the subset with the highest upper bound, regardless of where that subset is on the tree.*

**Advantages of Branch and Bound**

Regardless of the implementation specifics, B&B offers some significant advantages. The advantage of this procedure is that in using it one will be guaranteed to obtain the best solution in the region searched. This search region does not have to be of small scope to obtain the best solution either – this algorithm is the only one to guarantee a solution to the global optimization problem, as stated by (Hansen and Walster 2004):

> "The transcendent virtue of interval analysis is that it enables the solution of certain problems that cannot be solved by non-interval methods. The primary example is the

> global optimization problem...Even if interval procedures for this problem were slow, this fact cannot be considered a flaw. Fortunately, the procedures are quite efficient in most problems."

As this quote points out, finding a global optimal may seem a computationally expensive problem, and it would be if it is an explicit search. Instead B&B is an implicit search, which does not explicitly search all of the possible solution space; rather it uses information about a region to characterize all the solutions in that region. The regions are eliminated without ever explicitly searching the regions, thus the computations for an explicit search are reduced significantly when employing an implicit search.

An implicit search may seem questionable, being that is does not explicitly check everywhere in the solution space; however, as long as the characterization of regions is conservative then an implicit search will always work. Proof of this can be found in multiple sources (Moore 1966; Moore 1979; Kearfott and Kreinovich 1996; Hansen and Walster 2004; Montemanni, Gambardella et al. 2004). The internal consistency is apparent from the steps in the algorithm. If regions are conservatively characterized then the region with the global optimal is characterized such that it is not eliminated. Further elimination removes other regions, but the region with the global optimal remains.

Besides offering the advantage of finding the global optimal, the B&B algorithms can handle uncertainty in a formal manner. Since the performance of a subset includes all possible instances in that subset, B&B already includes uncertainty in the objective function over all instances in the subset. Since uncertainty is already included in the B&B algorithm, incorporating more uncertainty from different sources is possible. The algorithm would need to include this uncertainty in the interval bounds on each subset. This has already been used in optimization, where authors have applied the B&B Algorithm to uncertain information represented by intervals for multiple problems (Montemanni, Gambardella et al. 2004), (Yaman, Karasan et al. 2001; Karasan, Pinar et al. 2002). Researchers have even applied B&B under conditions of stochastic uncertainty (Norkin, Pflug et al. 1996). Because of the uncertainty that is already included in B&B algorithms, I believe that the B&B could be adapted to handle the uncertainty in design.

In addition, the steps of the algorithm are clearly defined individual processes with clear inputs and resulting outputs; this makes the process easily modular. Each step can be modified as seen necessary for the situation, and there is significant literature on how these steps can be modified for different situations (Lawler and Wood 1966; Moore 1979; Alefeld and Herberger 1983; Balakrishan, Boyd et al. 1991; Clausen and Traff 1991; Laursen 1993; Kearfott and Kreinovich 1996; Norkin, Pflug et al. 1996; Clausen 1997; Clausen and Perregaard 1997; Kearfott 1997; Adjiman, Androulakis et al. 1998; Adjiman, Androulakis et al. 1998; Clausen and Perregaard 1999; Montemanni and Gambardella 2002; Boyd, Ghosh et al. 2003; Hansen and Walster 2004; Montemanni, Gambardella et al. 2004). This makes B&B algorithms a good candidate for use as a framework for a design process.

Because of these assets of B&B, I use this algorithm as the basis for set-based design under uncertainty; this concept of a B&B approach to design is covered in Chapter 3. While this concept provides an overall research objective, it also provides a good context to view the elimination method given in Chapter 4. This elimination method extends rational decision making to include the interval uncertainty. In the next section, utility theory, the basis of rationality, is investigated along with another method of deciding under uncertainty, Information Gap Decision Theory.

### *2.4 Decision Methods for Design*

As pointed out in Chapter 1, the design process can be viewed as a series of decisions made to result in a final design (Mistree, Smith et al. 1990; Hazelrigg 1998; Chen 2001; Chen 2003). These decisions need to be made to progress toward the most preferred design. To make these decisions, one needs a means for comparing the decision alternatives, and a method for making the decisions. These items are reviewed in this section.

#### 2.4.1 Utility Theory: The Cardinal Approach to Preferences and Rational Decisions

In design, the designer needs to make decisions to progress toward the most preferred design. To do so, the designers need to make decisions based on their preferences formulated in a consistent, rational manner. Utility Theory helps in formulating these preferences in a mathematically rigorous manner.

Utility Theory provides a means to cardinally rank alternatives based on one's preferences. A cardinal ranking system provides the advantage of not only consistently expressing which alternative from a set is preferred, as is the case in an ordinal system, but also by how much. This distinction of the cardinal ranking system in Utility Theory is necessary for making decisions under uncertainty, where ordinal ranking systems may not be consistent.

The quantitative assessment of preferences in utility theory is performed in a manner that reflects, and is consistent with, the preferences of the designer; preferences, or the structural representation of those preferences, are not imposed on the designer. The benefits offered by Utility Theory make it useful in the design process, as well as this thesis. Utility Theory is used in this work to determine the actions of the designer under conditions of epistemic uncertainty.

This application of utility differs from the typical use; to show this approach to be valid an investigation into the basis of utility theory is needed.

## Basis of Utility Theory

The notion of utility grew out of the economic research community as a means for explaining the actions of an agent in an economy. Daniel Bernoulli (Bernoulli 1738) first introduced utility, saying that one's marginal utility decreases with increasing wealth. Economists use these ideas to explain markets and the action of agents in markets by explaining the behavior of agents as maximizing their utility. Applying utility in this manner, economists use indifference curves to express the preferences, and thus the actions, of an agent (Edgeworth 1881; Creedy 1986; Appleyard and Field Jr. 2001). This ordinal expression only allowed economists to say which alternative was preferred but not quantify by how much it was preferred. The ordinal ranking system works fine for comparing a small number of alternatives were the outcome of each alternative is certain; however, in a series of decisions, or uncertain conditions, require a more rigorous ranking system. A cardinal system of ranking meets these needs.

Utility as a cardinal measure of preferences was introduced by von Neumann and Morgenstern in their classic text *Theory of Games and Economic Behavior*. In this work they set out to define rational behavior of individuals in an economy, thus establish the foundation of economic activity. More specifically, the authors wanted to determine a set of mathematically rigorous principles to define how individuals ought to economically behave, or as the authors stated they wanted to determine "the mathematically complete principles which define 'rational behavior' for the participants in a social economy, and to derive from them the general characteristics of that behavior." (von Neumann and Morgenstern 1944)

To achieve their objective, von Neumann and Morgenstern needed to quantify the notion of preferences; to this end they argued that if one could determine their preferences between lottery alternatives with different probabilities then utility could be assigned to the alternatives based on the probability of that lottery. Consistent with this notion, von Neumann and Morgenstern determined axioms that serve as the basis of utility theory. These axioms are given in Figure 15.

Consider a system U of entities, a, b, c… Where a, b, c,… are abstract utilities for any number α and β, (0 < α < 1) and (0 < β < 1) satisfy the following axioms:

Axiom 1 a > b is a complete ordering of $U^2$.

This means: Write a < b when b > a. Then:

(1:a) For any two a, b one and only one of the three following relations holds:

a = b, a > b, a < b.

(1:b) $a > b, b > c \Rightarrow a > c$

Axiom 2 Ordering and combining.

(2:a) $a < b \Rightarrow a < \alpha a + (1-\alpha) b$

(2:b) $a > b \Rightarrow a > \alpha a + (1-\alpha) b$

(2:c) a < c < b implies the existence of an α with

$$\alpha a + (1-\alpha) b < c$$

(2:d) a > b > c implies the existence of an α with

$$\alpha a + (1-\alpha) b > c$$

Axiom 3 Algebra of combining.

(3:a) $\alpha a + (1-\alpha) b = (1-\alpha) b + \alpha a$

(3:b) $\alpha\left(\beta a + (1-\beta) b\right) + (1-\alpha) b = \gamma a + (1-\gamma) b$, where $\gamma = \alpha\beta$

*Figure 15: The Axioms of Utility, as stated by Von Neumann and Morgenstern (von Neumann and Morgenstern 1944)*

The importance of these axioms requires further explanation of each one:

1a. This is a completeness axiom stating that the preference of *a* to *b* can be determined.

1b. This axiom states the that preferences are transitive. If *a* is preferred to *b*, and *b* preferred over *c*, then *a* is preferred to *c*.

2ab. States that if *b* is preferred to *a* then even a chance at obtaining *b* is preferred to *a*

2cd. States that the utility function is continuous.

3a. States the utility and probability combinations are commutative in addition.

3b. States the utility and probability combination are commutative in multiplication.

These axioms serve as the rigorous framework of utility theory and form a positive affine transform. Because of this a quantitative utility can be assigned to alternatives thus making cardinal ranking possible. This advancement is crucial in the process of decision-making, as a cardinal ranking system offers the distinct advantage over an ordinal system in that it can be used under conditions of uncertainty.

With an ordinal ranking, a person can rank their preferences only when the outcomes are perfectly known; however when the outcomes are uncertain then the ranking is no longer sufficient. Without being able to quantify the preference of outcomes, making a decision under uncertainty is not possible. This concept is expressed in the following example. Suppose an individual has to choose between alternative *A* and alternative *B*. In the deterministic case, both the cardinal and ordinal systems work fine. Alternative *A* and *B* each have a single, perfectly known outcome. Selection is simple: if the outcome of *A* is preferred to outcome of *B* then select *A*. The results change under uncertainty.

Consider the case: alternative A has two possible outcomes *v* and *w*, where the probability of each outcome is 50%. Likewise alternative B has two possible outcomes *x* and *y*, where the probability of each outcome is 50%. Let *y* > *w* > *v* > *x*. In an ordinal ranking system one would not be able to choose between alternative *A* and alternative *B*; in order to make such a decision one would have determine *how much v* is preferred to *x* and how much *y* is preferred to *w* – in doing so one sets up a *cardinal* ranking system.

Since a designer knows how much one outcome is preferred to another, that designer can determine which alternative is preferable by applying the expected utility theorem. This expected utility theorem was developed by von Neumann and Morgenstern, who focused on characterizing the rationale for individuals in a social economy. To do so they first developed that rationale with respect to games because they argue that economic situations strongly resemble those in games: “We hope to establish satisfactorily, after developing a few plausible schematizations, that the typical problems of economic behavior become strictly identical with the mathematical notions of suitable games of strategy.” The games of strategy referred to were modeled as having probabilities of different outcomes (von Neumann and Morgenstern 1944). This probabilistic information led to the development of the expected utility rationale, whereas one selects the alternative with the highest expected utility as computed in the following equation:

$$E\left(U\right)=U(\left[p_1,q_1;p_2,q_2;...;p_n,q_n\right])=\sum_{i=1}^{n}p_iU\left(q_i\right)$$

where $p_1,p_2,...,p_i$ are the probabilities of outcomes $q_1,q_2,...,q_i$, respectively, and $E\left(U\right)$ is the expected utility from a specific decision. Because the designer must know the utility of the possible events, a cardinal measure of the designer’s preferences is necessary to take this approach. This expected utility theorem has been applied by many researchers (Luce and Raiffa 1957; Keeney and Raiffa 1993; Triantaphyllou 2000), who have made some significant improvements in the decision process.

Notable among these advancements, Keeney and Raiffa developed a method for eliciting one's preferences by performing series of preference tests on a lottery (Keeney and Raiffa 1993). This method is based on the assumption that each of the attributes is utility independent, so the utility function with respect to each attribute can be assessed independently. The essence of this process is diagrammed in Figure 16. In the process, one uses two values of an attribute, $y_1$ and $y_2$, to compare in a lottery with an alpha chance of $y_2$ and a 1-alpha chance of $y_1$. One then specifies the certain attribute value, $\hat{y}$, that one would consider equally as preferable as the lottery. One can then relate the utility of this certain value to the utility of the two attributes $y_1$ and $y_2$:

$$U\left(\hat{y}\right) = \alpha U\left(y_1\right) = \left(1-\alpha\right) U\left(y_2\right)$$

Using this process, the utility is assessed for different values of an attribute. A utility function can then be fit to these resulting utility values for that attribute. The utility functions for each of the attributes are then combined into a single, overall utility function using an additive or multiplicative form (Luce and Raiffa 1957; Keeney and Raiffa 1993; Triantaphyllou 2000).

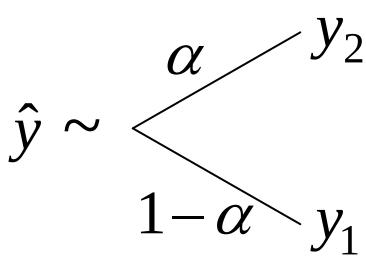



*Figure 16: As a step in assessing one's preferences with respect to an attribute, one determines the certain value of an attribute that one considers equally preferable to the lottery between two different values of that attribute.*

While utility theory is necessary for making rational decisions in conditions of probabilistic uncertainty, utility theory has application in cases where the information is not probabilistic. Peter Walley has applied rational decision making with imprecise probabilities (Walley 1991), and sparked a community of researchers to consider imprecise probabilities (http://ippserv.rug.ac.be/ 2005). In this thesis, I consider making decisions based on interval information. These intervals are obtained subjectively based on the belief of the designers and considered for elimination based on the deterministic rationale: $U(A) > U(B) \Rightarrow A > B$. This deterministic rationale is used to develop a rationale for interval information in Chapter 4 as part of the proposed elimination process used in Branch and Bound design. This proposed rationale is used in conjunction with the expression of preferences suggested by utility theory. Utility Theory has not been the basis of all decision methods. Next, the review is turned to an interesting departure from this typical approach to decision-making, and a new means of handling uncertainty in decisions.

### 2.4.2 Information Gap Decision Theory

The uncertainty models in probability theory and interval analysis represent information about the possible values of an uncertain parameter, and in turn, they require this information to be known about the uncertain parameter. However, this information can be absent in some cases. Without information about the uncertainty, a representation cannot be constructed, and the typical decision methods that use those representations no longer apply. Instead, one must turn to a decision method that does not require this information, such as Information-Gap (Info-Gap) Decision Theory (Ben-Haim 2001).

Info-Gap Decision Theory is a formal method that could be applied to make these decisions under extreme uncertainty. Due to this unique attribute, Info-Gap Decision Theory is investigated in this section, as a means for eliminating designs when no information is available about the uncertainty sources. Although Info-Gap is not used in this thesis, this work should be useful in future work on eliminating designs under extreme uncertainty.

Info-Gap Decision Theory is based on Info-Gap Models. An info-gap model states that the deviation from reality will be no larger than some threshold:

$$U(\alpha, \tilde{r}) = \left\{ r(x) : \left| r(x) - \tilde{r}(x) \right| \leq \alpha \right\}$$

where $U$ is the info-gap representation, $\tilde{r}$ is the nominal value predicted by some model, $r$ is that actual value, x is a vector of independent variables and $\alpha$ is the threshold. The uncertainty parameter represents the maximum deviation from the modeled nominal value; this info-gap model is an interval, of half-width $\alpha$, based around some predicted nominal value; the greater the value of $\alpha$, the larger the variation from reality. $\alpha$ is not to be confused with percentage or probability number; this value has no relation to the typical usage of $\alpha$ in probability and statistics. The info-gap model given in the above equation represents the interval of *possible* actual values of *r*:

$$[\tilde{r}-\alpha, \tilde{r}+\alpha]$$

This is one specific Info-Gap model. In the general case, an Info-Gap models is a parameterized set of nested intervals. Info-Gap models are a useful representation of uncertainty and the basis of the Info-Gap decision theory.

Based on these info-gap models Ben-Haim devised a means of decision-making that requires even less information than is present in an info-gap model (Ben-Haim 2001). These decision-making procedures, since they are based on a lack of information differ greatly from the previously examined decision-making methods.

Information Gap Decision Theory is actually a method of satisficing that is based primarily on an opportunity function and robustness function that are defined with respect to the uncertainty. Consider a system with an uncertain parameter z, which is modeled:

$$U(\alpha, \tilde{z}) = \{z : |z-\tilde{z}| \leq \alpha\}$$

where $\alpha$ is the threshold. The robustness function for that system is then defined as:

$$\hat{\alpha}(x) = \max_{z}\{\alpha : \text{minimal requirements are always satisfied}\}$$

where $\hat{\alpha}$ is the robustness function. To state this in mathematical terms, a performance function is defined:

$$f(x, z)$$

where $x$ is a vector that specifies the decision, $z$ is the factor that is uncertain, and $f$ is the performance measure of the system, in which a larger value is better. For a minimum acceptable level of performance $f_{min}$, the robustness function is as follows:

$$\hat{\alpha}\left(x, f_{\min}\right) = \max_{\alpha}\left\{\alpha : \min_{z \in U\left(\alpha, \tilde{z}\right)} f\left(x, z\right) \geq f_{\min}\right\}$$

The robustness of a decision is the maximum uncertainty in the factor $z$ such that the performance meets the minimum requirement, $f_{min}$. To illustrate the robustness function let’s say that the $x$ vector has been determined unambiguously, and there is not uncertainty in the model, so that the only uncertainty in the problem is from the parameter $z$. Since $z$ is the only parameter with uncertainty, the robustness is a function of z. Assume the level of satisfactory performance to be 8, and the performance function $f(x,z)$, with respect to z, behaves as given in Figure 17.

The robustness is the maximum deviation in $z$ from its nominal value such that this minimum performance requirement is still met. In this case $z$ could deviate 1.7 below the nominal value of $z$ and still meet the performance requirement but $z$ could only deviate 1.15 above the nominal value and still meet the performance requirement. The robustness of this particular design is therefore the 1.15, the minimum of the two maximums. This robustness value represents the maximum uncertainty that the system can experience such that the system still meets its requirements.

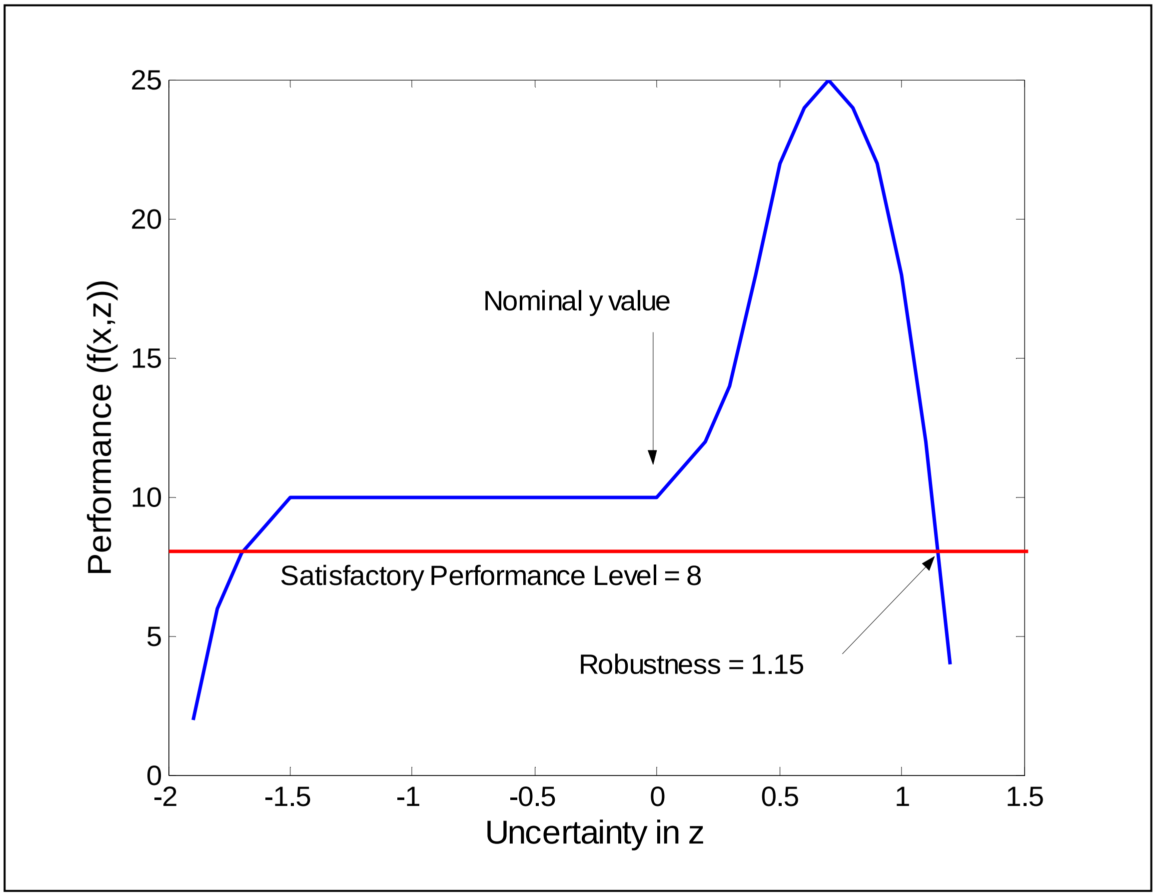



*Figure 17: Example of the Robustness function*

Just as there is a robustness function against failure in Info-Gap Decision Theory there is also an opportunity function. The opportunity function is based on the possibility of uncertainty working in favor of a system's performance and is the least uncertainty necessary for the system to have overwhelming success. The opportunity function is therefore defined as the minimum uncertainty necessary for the system to experience sweeping success. The opportunity function is then defined as:

$$\hat{\beta} = \min_{z} \{\alpha : \text{sweeping success is possible}\}$$

where $\alpha$ is the as in uncertainty threshold for achieving sweeping sucess, and $\hat{\beta}$ is the opportunity function. Stated in mathematical terms, this opportunity function, for a level of sweeping success $f_{big}$, is as follows:

$$\hat{\beta}\left(x, f_{big}\right) = \min_{\alpha}\left\{\alpha : \max_{z \in U\left(\alpha, \tilde{z}\right)} f\left(x, z\right) \geq f_{big}\right\}$$

The opportunity function therefore is the minimum level of uncertainty in the factor z such that the performance has the possibility of being as large as $f_{big}$. Unlike the robustness function, the opportunity function is not satisficing but is based on the possibility of exceptional performance. To illustrate the opportunity function the same example is used from the robustness function demonstration, where the vector *x* has been determined unambiguously. Assume the level of sweeping success would be a performance of $f\left(z\right) = 20$, and performance function $f\left(z\right)$, with respect to , behaves as given in Figure 18.

The opportunity is therefore the minimum uncertainty necessary in order to reach this level of exceptional performance. On the function plotted above the exceptional performance is reach with an uncertainty of 0.50 in the *z* parameter. 0.50 is thus the opportunity in this example. In considering the opportunity, a system that requires less uncertainty to have an opportunity at sweeping success is preferred to a system that requires larger uncertainty for an opportunity at sweeping success.

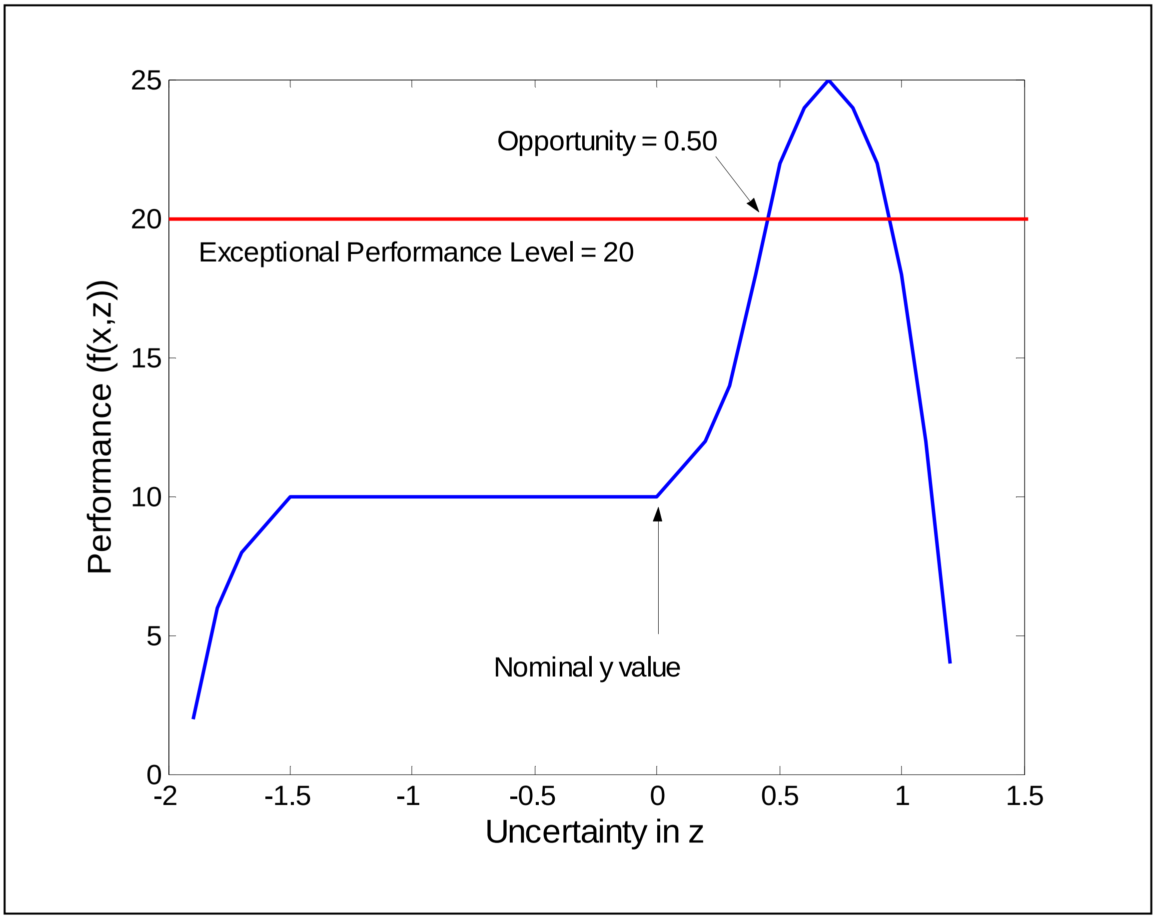



*Figure 18: An Example of the Opportunity Function*

The opportunity and robustness functions are used in info-gap decision theory to make decisions that minimize the exposure to failure and maximize the exposure to exceptional success. This is the crux of Info-Gap Decision Theory:

***Principle of Info-Gap:*** One should maximize the uncertainty needed to make a system unacceptable and one should minimize the uncertainty needed to enable sweeping success.

Although the exact formulation varies, this principle remains at the heart of the Info-Gap Decision Theory. Decisions based on this theory are satisficing in nature. That is the primary concern when deciding is that a minimum level of performance, or satisfactory performance is met. This idea was introduced by Herbert Simon (Simon 1996). There are other satisficing methods for decision-making (Stirling 2003), but none that have employed these ideas of robustness and opportunity in this manner.

Info-Gap Decision methods are most useful in engineering design when a decision has to be made unambiguously without reducing the uncertainty. Info-Gap is especially useful when the model of performance is far more accurate than the model of the uncertainty in parameters. In this instance, one could be sure of how the system would react to variation in parameters but not sure what are those variations. Although these methods do have their place in engineering design, the theory does have some serious shortcomings when applied to design.

One of the main shortcomings with this decision-making process for design: it is based on the outcomes of extreme failure and extreme success. Deciding based on these extremes can be very useful when the decision involves extreme uncertainty that cannot be reduced significantly and a decision has to be made. In most of engineering design the uncertainty can be significantly reduced by further developing or prototyping the design; designers do not have to unambiguously decide on a specific design without reducing the uncertainty. In reducing this uncertainty the design can be made to take advantage of the opportunity instead of just hoping uncertainty works in the design’s favor.

Another main shortcoming of Info-Gap Decision Theory is that it does not cover cases involving more than one source of uncertainty, which are prevalent in design. Nor does it cover cases in which enough information about the uncertainty is known to construct some sort of representation. For these shortcomings the Information-Gap Decision Theory will not be used, however if one could amend info-gap to incorporate these needs, then it could be extremely useful. For this reason, Info-Gap should be considered in future work on eliminating or deciding on designs.

## *2.5 Summary of Background*

In this chapter, I reviewed research in uncertainty, design, optimization, and decision-making. Within each section, I pointed out the valuable aspects of the work that I will drawn from in my contribution, as well as noted the shortcoming, where I could contribute. These valuable aspects and opportunities for improvement are pointed in Table 6.

**Table 6: Value and Opportunity in Reviewed Research**

| B&B Design Step | Value in this Thesis | Opportunity for Improvement |
|---|---|---|
| Interval Analysis | Interval representation of uncertainty<br>Interval arithmetic<br>Error bounds on model uncertainty | Applying this representation and analysis to epistemic uncertainty and the design problem |
| Probability Representation of Uncertainty | Monte Carlo Simulation sample approach to simulation<br>Common random numbers used in Monte Carlo | Applying the concept behind these simulation techniques beyond stochastic uncertainty, to intervals |

| | | |
|---|---|---|
| Pahl and Bietz Design Methodology | Systematic approach to design with organized phases and steps | Needs to integrate the validation process throughout the design process<br>Needs to account for uncertainty<br>Needs internally consistent steps that converge toward the most preferred design |
| Vee Model of Systems Engineering | Formal method for validating designs setup in each step | Account for uncertainty<br>Internal consistency in steps that converge toward the most preferred design |
| Set-Based Concurrent Engineering | Accounts for uncertainty and concurrency | Needs to be formalized into a systematic approach with organized steps |
| Robust Concept Exploration Methods | Accounts for uncertainty<br>Internally consistent steps that converge (not necessarily on the most preferred design) | Formal means of validating designs throughout the process |
| Branch and Bound Algorithm | Rigorous steps supported by mathematics<br>Theoretical Convergence Guaranteed<br>Structure to formally handle uncertainty | Solution space is well-defined; not so in design<br>Design is concurrent on different subsystems – this is not an issue in B&B<br>Uncertainty from multiple sources<br>Design is very cost/time influenced (these challenges are expanded on in Chapter 3) |

| | | |
|---|---|---|
| Utility Theory | Mathematically sound representation of one's preferences<br>Necessary for making decisions under uncertainty<br>Applied under epistemic uncertainty in thesis | Applied only in stochastic uncertainty or deterministically<br><br>Need to be extended to intervals |
| Information-Gap Decision Theory | Useful in Future Work<br>Valuable for making decisions under severe uncertainty | Application when uncertainty is represented<br>Application with multiple sources of uncertainty<br>Based only on extreme outcomes |

As pointed out in Table 6, there is a need for a design process that formally handles the problem of design under uncertainty. For this, I draw from the work in Set-Based Concurrent Engineering and the Branch and Bound (B&B) Algorithm. The knowledge from the B&B Algorithms are used as a concept for a formal approach to set-based design; this concept is argued for and presented in Chapter 3. Before moving on to this chapter, the reader's attention is returned to the thesis map in Figure 19 to see what has been presented and what will be presented.

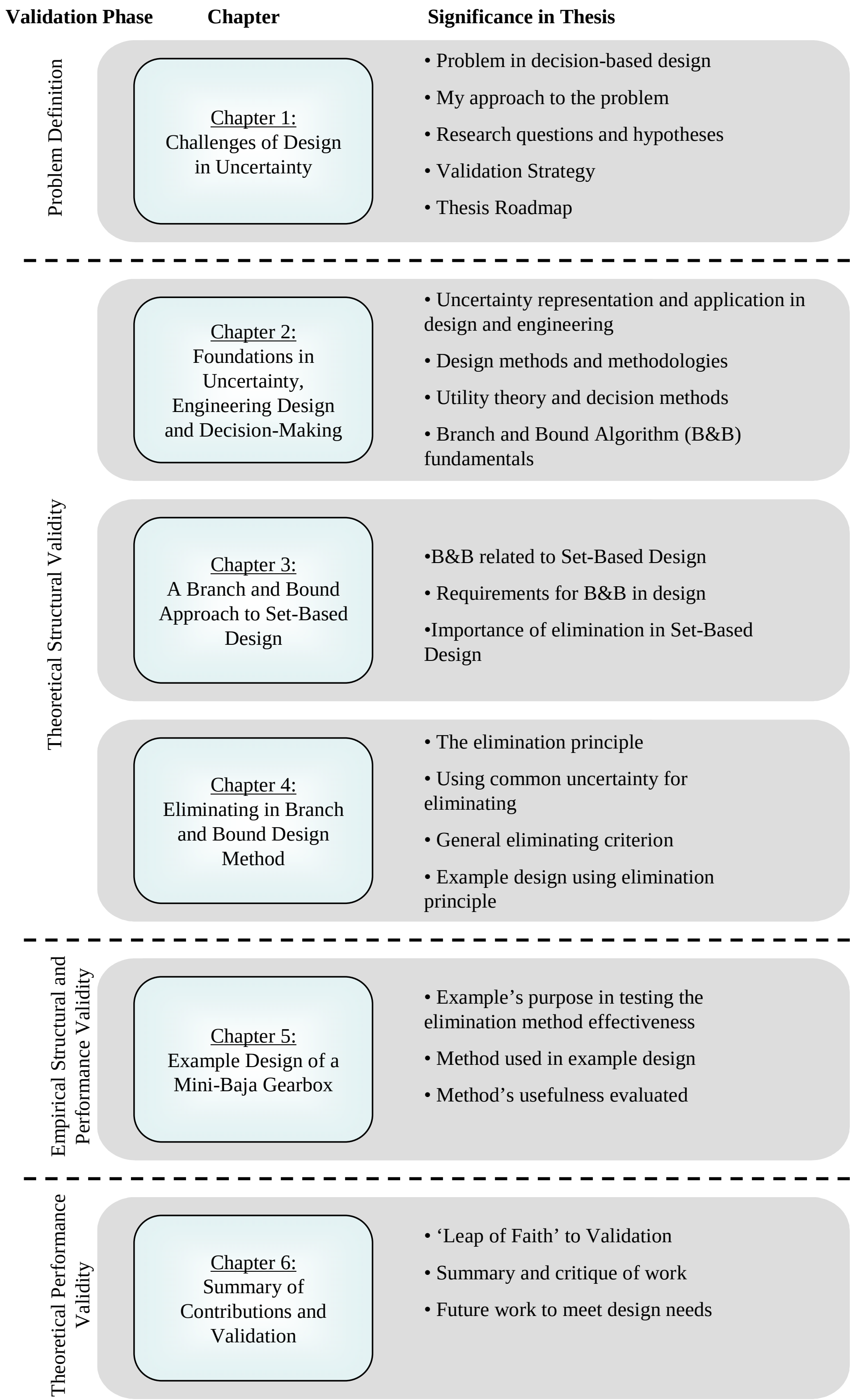



*Figure 19: Thesis Roadmap*

# CHAPTER 3

# A BRANCH AND BOUND APPROACH TO SET-BASED DESIGN

In Section 2.3, design methods were reviewed in search of formal approach to designing under uncertainty. While each design method is useful in design, I believe that none of these methods are sufficiently formal for design under uncertainty. In this chapter, I present a concept for a formal approach to set-based design under uncertainty. This concept provides both the motivation and the context for the elimination method.

This concept is inspired by the Set-Based Concurrent Engineering (SBCE) used at Toyota (Ward, Liker et al. 1995; Sobek and Ward 1996; Sobek II 2004). This set-based approach has some significant advantages over the traditional, point-based approaches; these advantages are expressed in Section 3.1. Although SBCE has some significant advantages over traditional design, it has not been formalized. I believe the Branch and Bound (B&B) Algorithm is similar enough to SBCE to allow an easy mapping between the two, while the algorithm's explicit steps and tests provide the formality desired in a formal set-based design method. These advantages are examined more in detail in Section 3.2. In Section 3.3, I examine the challenges of creating a B&B approach. Then in Section 3.4, I present my concept of the B&B design process and express how the formality in B&B can be translated to design – I specify the requirements I believe to be necessary and the objectives the designer should keep in mind. These requirements and objectives then are summarized in Section 3.5 and the thesis roadmap is revisited.

### *3.1 The Benefit of a Set-Based Approach to Design*

Among the design methods reviewed in Section 2.3, SBCE was the most promising in handling the challenges posed by uncertainty in design. The advantages of SBCE over traditional, point-based, design approaches are recognized by multiple researchers (Chang 1994; Ward, Liker et al. 1995; Liker, Sobek II et al. 1996; Sobek and Ward 1996; Finch 1997; Finch 1997; Parunak, Ward et al. 1997; Wu 1999; de Weck 2002; Costa and Sobek II 2003; Ford and Sobek II 2004; Rekuc and Paredis 2005). The reasons for these advantages are as follows:

1. **A set-based approach handles uncertainty by encouraging learning.** In traditional design approach, if uncertainty is considered, it is considered via safety factors and is not a major factor in design decisions. The designers make their decisions based on incomplete information. Instead in SBCE, if the designers face uncertainty that inhibits their ability to decide, they learn more about the decision alternatives (Ward, Liker et al. 1995; Sobek and Ward 1996; Sobek II 2004). This is particularly true about decisions early in the design process; rather than commit to poor-performing alternatives, the designers explore the alternatives further (Ward, Liker et al. 1995). In this way, the designers are decreasing uncertainty by learning more.
2. **A set-based approach to elimination allows one to converge on the most preferred design.** Despite the uncertainty in design performance, designers, using traditional design methods, typically eliminate all but one alternative in their decisions. This just exacerbates the problems caused by ignoring uncertainty. Even if the designers consider uncertainty, it is unlikely they could find the most preferred design via a sequence of point decisions. This is because all the decisions in the design process are coupled: the

decision the designer makes in one decision is dependant on the result of the other decisions. To determine the specific alternative in the first decision that leads to the most preferred design, the designer would need to know the outcomes of all future decisions. Since the designer does not know the outcomes of all future decisions, the designer could not select the single alternative in each decision that leads to the most preferred design.

SBCE avoids this problem by not forcing the designers to make point decisions; instead, the designers decide on a set of designs. By this approach, the designers could converge on the most preferred design. In addition, designers using SBCE explore many concepts in more depth than traditional designs methods. Because of this practice, the designers have the opportunity to find better solutions based on radically new concepts. Developing more conservative, proven concepts in parallel provides safety in the process: if the radical idea does not work out, the conservative option is there (Ward, Liker et al. 1995; Sobek and Ward 1996).

3. **A Set-based approach is better suited for concurrent engineering.** First, the designers communicate in terms of sets of feasible designs. This allows designers to work concurrently while knowing exactly what the other designers are considering. Since designers are considering sets of possibilities, the designers can consider multiple combinations of subsystems, giving them a better chance of finding the most preferred design. Secondly, since designers are already considering all the possibilities in the set, design changes are communicated with less effort because the different designers know the changes possible in the set (Ward, Liker et al. 1995; Sobek and Ward 1996).

   This concurrent approach also makes sense when considering the impact of uncertainty. These designers investigate their set of potential solutions, decreasing the uncertainty about those solutions, and then sharing the findings with the other designers. By not jumping to decisions, these teams should be able to make more informed decisions in the end and produce better designs.

4. **Set-based approach promotes designer learning.** The above advantages offered in a single design process, if realized, would be enough to merit one to implement a set-based approach, but this approach offers an additional benefit over successive, similar design problems. Since SBCE requires designers to search multiple branches in the solution space, the designers learn about the tradeoffs encountered in different regions of the solution space. When solving a similar design problem in the future, the designer can apply this prior knowledge about the design space; the designer knows the performance available from each branch based on the previous searches. The designer will therefore spend less time characterizing the performance of these branches and more time ideating and trying out new branches. This should result is a design process where more ideas are considered and the designer moves more efficiently. Toyota actually enforces this learning process by means of lessoned learned books. Engineers update these books during the design process to reflect lessons they have learned during the process (Ward, Liker et al. 1995; Sobek and Ward 1996).

While a set-based approach to design offers these advantages over the traditional point-based approaches, the approach still has room for improvement. Namely, the approach lacks specific steps and phases, which can result in an efficiency loss when designers waste their efforts on unproductive steps and become confused. The advantages of formulating these steps based on the Branch and Bound algorithm are given in the next section.

## *3.2 The Benefit of a Branch and Bound Approach to Set-Based Design*

In the previous section, the benefits of the set-based concurrent engineering (SBCE) approach were highlighted. These advantages are discussed in more depth by Ward and Sobek (Ward, Liker et al. 1995; Sobek and Ward 1996; Sobek II 2004). Although these advantages of the SBCE have been observed, there is no formal systematic process to obtain these benefits; there is no set-based design methodology. The advantages of a systematic design methodology were argued for by Pahl and Beitz (Pahl and Beitz 1996) and also apply to a set-based design approach.

I draw the structure for a design methodology from the Branch and Bound (B&B) Algorithm. This algorithm has been applied successfully in optimization and computer searches (Lawler and Wood 1966; Alefeld and Herberger 1983; Balakrishan, Boyd et al. 1991; Clausen and Traff 1991; Laursen 1993; Norkin, Pflug et al. 1996; Clausen 1997; Clausen and Perregaard 1997; Clausen and Perregaard 1999; Boyd, Ghosh et al. 2003; Montemanni, Gambardella et al. 2004), and I believe it has significant potential benefits to merit applying it to SBCE. First, this formal approach provides set-based designers with a framework to organize the process and specific steps to execute, eliminating many of the wasteful actions a designer may take, as well as keeping the designers from reinventing the design process for each design problem. Second, this approach offers the rigor missing in SBD: one knows that the most preferred solution is in the set of remaining designs and that the process will converge on the most preferred solution. Third, this approach requires the designers to make decisions that reflect the extent of their knowledge. Lastly, B&B is very similar to SBD – B&B and SBD shared the same overall objective and are similar in approach – thus, applying B&B is merely an extension of the existing algorithm to a new application. In this section, I articulate these potential benefits.

**The advantages of a Branch and Bound Approach to design**

A B&B approach to design holds some significant advantages over the standard set-based approach thanks to the rigor offered in a B&B approach. Namely, I see the following advantages:

1. **A systematic B&B approach should reduce wasteful actions in the design process.** This benefit of a systematic process is not specific to a B&B design approach, rather the value is realized in implementing any systematic approach. This is recognized by Pahl and Beitz: “Systematic procedures help to render designing comprehensible and also enable the subject to be taught.” Pahl and Beitz then continue, “Systematic procedures merely try to steer the efforts of designers from unconscious to the conscious and more purposeful paths (p. 11 in (Pahl and Beitz 1996)).” Similarly, a systematic approach based on B&B, would spell out the actions for the designers to take explicitly. This should eliminate any potential wasteful actions and guide the designers, step by step, to find the most preferred design. Because the steps are explicit, the designers don’t have to reinvent the design process; the designers do not waste any effort determining their next step – they know their next step. This allows the designers to focus on creating potential solutions and on finding the most preferred of those solutions. This benefit was recognized by Pahl and Beitz (p. 11 in (Pahl and Beitz 1996)).

This first benefit could be realized by using any systematic design approach. A systematic method can help guide a designer through the steps of the design process, making the designer less confused and more purposeful in action. This makes the design process more efficient than without a systematic method; the design is completed more quickly and with fewer resources expended. However, this does not mean that the steps of the process cannot be formulated to improve efficiency. Well formulated and thought-out steps in the design process can lead the designer to the most preferred solution in less time and with fewer resources. I believe a more efficient design process can be modeled after the B&B Algorithm. I believe such a process should have the following additional advantages:

2. **The approach should efficiently eliminate dominated designs and converge on the most preferred solution.** B&B algorithms and SBCE are both set-based methods of elimination to find the solution. In B&B algorithms, the search space is divided into smaller sets. Each of those sets is characterized and the inferior or infeasible sets are eliminated, while the remaining sets are divided again. The crux of SBCE is recognized by Ward: “Toyota designers think about sets of design alternatives, rather than pursuing one alternative iteratively. They gradually narrow the sets until they come to the final solution.” (Ward, Liker et al. 1995) Both B&B algorithms and SBCE are a search by process of elimination, in which impossible or inferior solutions are eliminated until a sufficiently small number of solutions remain.

   However, the B&B algorithm is more rigorous and more efficient in execution. The B&B algorithm efficiently reduces sets – all sets that can be eliminated, are eliminated, while no set that could contain the solution is eliminated. This means that in using the B&B, one would not risk eliminating the most preferred design while one should converge on the most preferred solution (Clausen and Traff 1991; Clausen 1997; Hansen and Walster 2004); this elimination is as efficient as possible with the given information. If the same rigorous elimination could be applied to design, this efficiency could be realized as well.

Although efficient elimination allows one to converge faster on the most preferred solution, elimination is not the only factor in efficiently finding a solution. In the B&B Algorithm, the search is organized efficiently. The sets are created in branches that allow more elimination. In addition, B&B has efficient means of searching the subsets (branches) in the iterative process. If this organization could be implemented in SBCE, then the search for the most preferred design would be more efficient. Researchers have been developing and evaluating different search methods for B&B algorithms (Clausen and Perregaard 1999), some of which are reviewed in Section 2.3. It would be beneficial to build on this knowledge in formulating a design process.

3. **B&B algorithms are similar to set-based design, allowing knowledge to be drawn from B&B algorithms for set-based design.** As pointed out in the previous step, there are strong similarities between the SBCE and B&B algorithm. Because of the similarity between B&B algorithms and set-based design, one could draw from the knowledge about the B&B algorithms in creating and improving the B&B design process.

The advantages that I foresee by a Branch and Bound approach to design, both over other SBCE and traditional design approaches are significant enough to motivate the realization of such an approach. However, such an approach cannot be created overnight; there are significant challenges that stand in the way of creating a successful B&B approach. These challenges are presented in the next section.

### *3.3 Challenges of a Branch and Bound Approach to Set-Based Design*

In the previous section, the possible advantages of a Branch and Bound (B&B) approach to design were highlighted; among these advantages, it was recognized that realizing the approach would be eased by the prior work in B&B; however, realizing the approach is not easy. There is some difficulty transferring the knowledge about an algorithm to the process of design. The possible challenges that I foresee in realizing a B&B design approach are as follows:

1. **The design space is not as well defined as it is in optimization.** In optimization the search space is clearly defined in terms of variables and alternatives from the start of the search (Nemhauser, Rinnooy Kan et al. 1989; Clausen 1997; Kearfott 1997; Boyd, Ghosh et al. 2003). However when beginning a design, the designers do not have a well-defined design space. Many of the design variables or alternatives in the design details remain unknown until these subsequent decisions are reached. The designer may have difficulty branching over such a design space. Additionally, the designer may have difficulty in bounding the design performance when they do not know the future aspects of the design that effect design performance.
2. **Modern design is concurrent.** As the project size and complexity of designs has increased, the design process has involved an increasing amount of concurrent engineering (Hoffman 1998). Concurrent engineering is based on the idea of near decomposability, as introduced by Herbert Simon (Simon 1996); in short, the interaction between subsystems is weaker than the interaction within a particular subsystem. Based on the assumption of near decomposability, the system being designed is decomposed into subsystems with methods such as the Vee Model of Systems Engineering (Forsberg

and Mooz 1992). Then designs teams work in parallel on these subsystems before the subsystems are integrated for the final design. While B&B algorithms have been created that work in parallel, all of these relate to searching different branches for the best system (Clausen and Traff 1991; Laursen 1993; Clausen and Perregaard 1997); none relate to decomposing a larger system into parallel searches for the different sub-systems of the same solution. In applying a B&B design process, one must determine how to decompose the subsystems and how the search for the different subsystems should be conducted in parallel. While, a set-based design approach, such as a B&B approach, may be better suited to handle the concurrent challenge than a traditional design methodology, this is not included in the Branch and Bound algorithm and needs to be addressed.

3. **Uncertainty is ubiquitous in design, but generally not addressed in B&B algorithms.** Uncertainty crops up from many sources in design (Aughenbaugh 2004). Whether from future design decisions, environmental conditions, or model inaccuracy, this uncertainty complicates the design process. For example, designers cannot be sure of design performance, and thus have a difficulty deciding. While uncertainty has been incorporated in some B&B implementations (Norkin, Pflug et al. 1996; Montemanni, Gambardella et al. 2004), it has only been considered for specific sources of uncertainty, applied in specific parts of the algorithm. There is not a general B&B algorithm that incorporates the possible sources and types of uncertainty encountered in design. The uncertainty must be recognized and incorporated in any rational process (p. 1 in (Walley 1991), including design (Aughenbaugh 2004).

4. **The objective incorporates both the product performance and the process performance.** B&B algorithms are executed more swiftly and at less cost than a design

process (Nemhauser, Rinnooy Kan et al. 1989; Clausen and Traff 1991; Clausen 1997; Clausen and Perregaard 1999; Boyd, Ghosh et al. 2003), therefore these algorithms use an objective function that only incorporates the performance of the solution. However, in design, the designers and their managers are not only concerned with performance of the product but also the cost of the design process. This cost needs to be considered in the objective function when searching for the most preferred solution. Thus, in the design process, the designers are not just searching for the most preferred solution, but the most preferred solution and process. This may lead the designers to not investigate an additional branch, even if they know the most preferred design could be on that branch, due to the cost of that additional search.

5. **Economic feasibility of the method and its implementation.** To the best of my knowledge, nobody has implemented a B&B design approach, therefore it is unknown whether such an approach is economically feasible. First, it is not known if this approach can converge with reasonable time and resources expended. Although theoretical convergence could be guaranteed in the approach, one could not guarantee that convergence to be economically feasible without implementing the method. The method needs to be formulated to be efficient at converging.

    Not only would a B&B method have to be economically feasible in its working, but if the approach is ever to extend beyond an idea then it would also have to be culturally acceptable. The engineers must be taught the process, and the tools needed in the process must be readily available. Although Toyota has implemented a similar, set-based method, they did so over the course of 20 years (Ward, Liker et al. 1995; Sobek and Ward 1996; Sobek II 2004). Implementing a B&B method at any company, over a short period of time, requires the implementation to be well thought out.

I believe these challenges need to be addressed for a usable B&B design process to be realized. How I believe these challenges should be addressed is the focus of the next section, as I present my concept for a B&B approach to design followed by requirements to realize this approach.

## *3.4 Branch and Bound Approach to Design*

As pointed out in Section 3.2, a Branch and Bound (B&B) approach to design should offer some significant advantages. Because of this, I present my concept for a B&B approach to design in the next subsection, followed in subsection 3.4.2 by the requirements I see necessary to formalize this approach.

### 3.4.1 My Concept of a Branch and Bound Design Approach

The Branch and Bound Algorithm is a formal optimization method that closes in on the most preferred solution by rigorously evaluating and eliminating designs algorithm with the guarantee that the most preferred solution is not eliminated (p. 11-12 in (Hansen and Walster 2004)) (Boyd, Ghosh et al. 2003). Applying that same rigor to set-based design is the underlying concept for my Branch and Bound approach to design.

The steps of the B&B Algorithm provide a rigorous method for closing in on the most-preferred solution. The similar, rigorous steps are applied in my design approach. My concept for these steps is shown in *Figure 2* alongside the steps of the B&B Algorithm (Nemhauser, Rinnooy Kan et al. 1989; Clausen and Traff 1991; Clausen 1997); one should notice the steps in both processes are very similar.

| **B&B Algorithm** | **B&B Design Process** |
|---|---|
| Step 1: Branch solution space into groups | Step 1: Formulate decision – group designs |
| Step 2: Bound performance of groups (branches) | Step 2: Bound performance of groups (branches) |
| Step 3: Eliminate branches based on bounds | Step 3: Eliminate designs based on bounds |
| Step 4: Select remaining branch to search in next iteration | Step 4: Select remaining group to search in next iteration |
| Repeat with next branch | Repeat with next decision |

*Figure.20: Steps in the B&B Algorithm and the B&B Design Process are very similar*

To begin an iteration of my B&B design approach, the designer selects the variables or alternatives in the design to decide on. These variables or alternatives represent all remaining designs in the feasible set that could be created with those alternatives or variables values. Thus, in setting up the decision, the designs are grouped into families that correspond to the alternatives or variables in the decision, which is equivalent to dividing (branching) the solution space in the B&B Algorithm. While some guidelines could be given for selecting the variables to use in the decision, this step requires the creativity of the designers to generate the alternatives and how the variables used will define the design.

In step two, the performance of the families created in step one are bounded. This bounding is performed with computational models or physical prototype testing and includes all members of the family. Additionally, these bounds need to include the uncertainty from both the model(s) and the environment; therefore the bounds represent all the possible performance of the families. Likewise, in the B&B algorithm, the subsets are bound by computational model. However, the bounds computed typically do not include uncertainty, except for a few exceptions that consider it from a single source  (Norkin, Pflug et al. 1996; Montemanni, Gambardella et al. 2004).

In my third step, the performance bounds of the families are used to eliminate the dominated families. A dominated family cannot perform as well as at least one other family. These dominated families cannot contain the most preferred design, thus there is no reason to consider these dominated subsets further. This step is similar to pruning in the B&B algorithm. However, in some B&B algorithms this is step is not nearly as important as it is in design. In these algorithms, the search begins with the optimal performance as a known value; in other words, it is known exactly what is searched for. This allows such algorithms to terminate when they find a design that meets that performance (Lawler and Wood 1966; Nemhauser, Rinnooy Kan et al. 1989; Belegundu and Chandrupatla 1999). Design is different.

One does not begin the design process knowing the possible performance of the designs. Since they do not know how good performance of the designs can be they do not know for sure that a design is the most preferred until they can establish that design dominates all other designs. Thus, elimination is crucial in the design process in finding the most preferred design. Since elimination is crucial to the success of a B&B design process, in addition to other set-based design approaches, the focus of this thesis and Chapter 4 is on realizing an effective elimination method.

The elimination method is formulated to eliminate as many design alternatives as possible without eliminating the most preferred. While this elimination approach allows the design process to converge faster on a particular design solution, the approach may seem in contrast to the set-based approach at Toyota, where they intentionally delayed design decisions (Ward, Liker et al. 1995; Sobek and Ward 1996). This is because Toyota does not explicitly account for all of the uncertainty and bound the performance of the possible designs, instead Toyota recognizes that the uncertainty implicitly. Their engineers are encourages to delay reducing the set until they can be sure of the alternative they are selecting. This is the same approach taken in this thesis: only eliminate designs alternatives when that is supported by the information available.

After the elimination step, my processes iterates, but it must be decided which subset will be iterated on. For this, a search strategy uses the bounds from the characterization to determine which subset(s) should be investigated further. My process continues in this fashion until the most preferred solution is reached or until the results are sufficiently close to the most preferred.

These steps of my B&B design process are tied together in Figure.21, which shows the purpose of the overall steps in the method’s iteration. As the reader can see, from both Figure 3.1 and 3.2, these steps of the B&B design approach that I am advocating are similar to the steps of the B&B Algorithm while still specific to the design process.

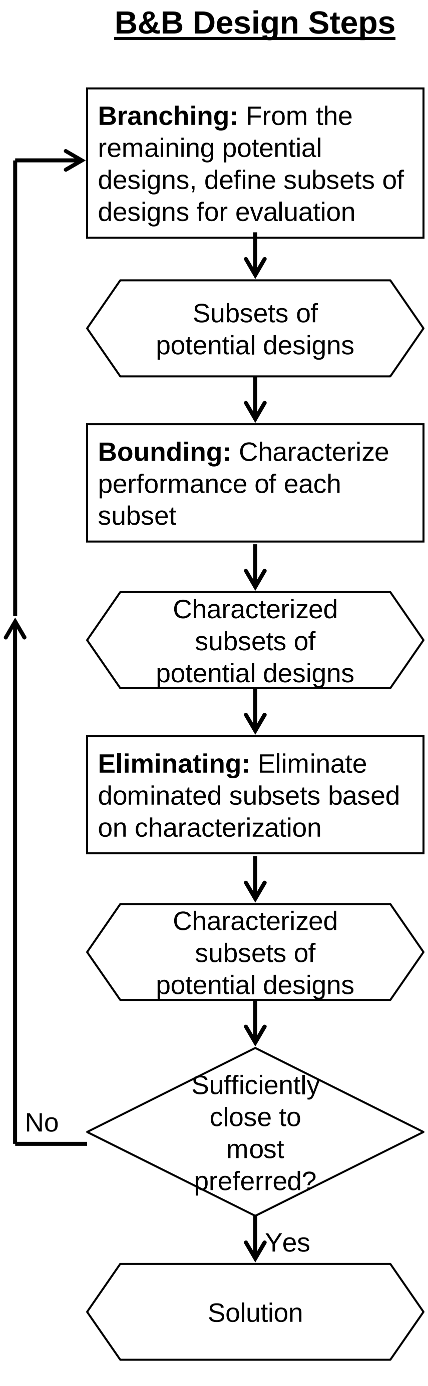



Figure.21: Steps in the B&B Design Approach

While the designers do need to understand the purpose behind the steps and the structure of the overall method, that alone does not provide a rigorous approach to design. Methods for each of these steps need to be explicitly spelled out for the designers to use. In the next subsection, requirements for these methods are given for each step in the design process.

### 3.4.2 My Requirements for a Branch and Bound Approach to Design

I believe a Branch and Bound (B&B) approach to design offers some significant advantages, but just like the B&B algorithm, these advantages can only be realized fully if some basic requirements are met. The requirements for the B&B algorithm are given by multiple authors (Nemhauser, Rinnooy Kan et al. 1989; Kearfott and Kreinovich 1996; Clausen 1997; Hansen and Walster 2004). Since requirements like these do not exist for a B&B design approach, I have formulated my requirements for a B&B design approach based on the B&B algorithm requirements and additional requirements I foresee in design. I see these requirements as necessary to keep the rigor of the process – avoid eliminating the most preferred design while searching all of the design space and remaining internally consistent. My requirements are presented in this section in a step by step manner.

In addition to these requirements, the designer should strive to meet some objectives in each step to improve the efficiency of the design process – converging quickly on the most preferred design with minimum resource expended. The rigor of the process is not lost if these objectives are ignored; however these individual objectives should be considered so that the B&B approach arrives at a design solution quickly and at reasonable cost. These objectives are often the focus of the research for those in the B&B algorithm community (Kearfott 1997; Adjiman, Androulakis et al. 1998; Adjiman, Androulakis et al. 1998) (Clausen and Traff 1991; Clausen and Perregaard 1997; Clausen and Perregaard 1999). For example, some researchers have investigated the efficiency of different search strategies under different circumstances (Adjiman, Androulakis et al. 1998; Adjiman, Androulakis et al. 1998; Clausen and Perregaard 1999). In a similar manner, the objectives in a B&B design approach should be the focus of improvement by researchers. Based on the work in B&B and the objectives in the design process, I have formulated what I believe to be the objective for each step in the process. Next, I present what I believe to be the requirements and objectives for each step in the process.

### Branching in B&B

The purpose of the branching step is to divide up the remaining set of potential designs into subsets to be evaluated. To ensure that the design process is internally consistent, the branching step must create subsets such that they include all designs without adding any designs that were considered in the original set. These high-level requirements can be decomposed into the requirements that I see for the branching step:

1. **The branches must contain the feasible set.** Designers create branches based on the designs remaining in the feasible set, but it is possible to accidentally eliminate designs by not including a branch to parts of the feasible set. When creating the branches, the designer must not eliminate any designs; that is not the purpose of this step and it ruins the rigor of the process. This requirement does not mean that every possible design needs to be included, as the bounded rationality of the designers makes this impossible (Simon 1982). Only the designs considered in the feasible set need to be included in the branches.
2. **Each branch must be less inclusive than the feasible set.** In optimization, convergence is ensured if the size of each generated subset is smaller than the original set (Clausen 1997; Boyd, Ghosh et al. 2003). The same requirement holds true for design. The method

cannot converge if the subset considered at each branch is larger than the parent set by inadvertently adding designs to the set considered at each step. This requirement needs to be balanced with the next.

3. **Branching must foster the creative nature of design.** Based on requirement 2, one could not allow new design ideas to be considered, thus stifling creativity, but this should be avoided. Pahl and Beitz recognized the importance of creativity and fostering that creativity in the design process, as creativity generates new ideas and novel combinations of existing ideas; these are vital for significant advancements in design (p. 11, 50 in (Pahl and Beitz 1996)). The creativity of the designers cannot be lost in the rigorous of the B&B design approach. Creativity must be part of the branching process, where innovative ideas can be added as seen fit. Of course, this need to include these ideas must be balanced with the need to converge.

I see these requirements necessary for the branching step to be rigorous and internally consistent; however, the process may perform terribly. For better performance, the one needs to consider some objectives in the branching step that lead to the elimination of more dominated designs at a lower cost. The objective that I see is as follows:

- **Branching should create subsets that allow the most elimination possible with the least resources.** With B&B Algorithms, empirical evidence has shown that the choice of branches can be very important to the run time of the algorithm (p. 304 in (Nemhauser, Rinnooy Kan et al. 1989)). Similarly, the branches a designer chooses can have a significant impact the efficiency of the design process. While designers must create sets that can be characterized, the design process is most efficient if these sets are created such that the maximum amount of feasible set is eliminated. This is a particularly difficult objective to formulate given that it is based on future steps in the design process and the size of a set is not easily quantified.

My requirements and objectives in branching have a strong dependence on bounding and eliminating, both future steps in the design process. Therefore these future steps must be formulated more rigorously before that same rigor can be applied to the branching step. Further progress in the branching step will be left as future work, while the focus is shifted to the bounding and eliminating steps.

### Bounding in B&B

The purpose of bounding is to characterize the subsets created in the branching step so that the inferior subsets can be eliminated. Because this characterization is used to determine the designers' action in future steps, if the characterization is inappropriate then those future steps suffer. If the characterization results in bounds that do not truly bound all of the possibilities then one may eliminate the most preferred solution, whereas unnecessarily large bounds (overly-conservative) will inhibit elimination and slow convergence. Bounding is equally important in the B&B algorithm, as recognized by Clausen: "The bounding function is the key component of any B&B algorithm in the sense that a low quality bounding function cannot be compensated for through good choices of branching and selection strategy." (Clausen 1997) The importance of bounding in B&B has led many authors of B&B process to introduce the bounding first (Moore 1966; Moore 1979; Alefeld and Herberger 1983; Kearfott and Kreinovich 1996; Hansen and Walster 2004), as each sees bounding as the foundation of B&B.

To avoid eliminating a design that could potentially be the most preferred, one needs to characterize the subset performance with conservative bounds. Conservative should not be confused with an underestimate of performance – because it is not. Rather, a conservative characterization of performance is one which definitely contains the true performance: if the conservative characterization of performance is a set of possible performance $U$ and the true performance is $u$ then $u \in U$. This conservative characterization is required in each B&B procedures (Lawler and Wood 1966; Moore 1979; Alefeld and Herberger 1983; Clausen and Traff 1991; Laursen 1993; Kearfott and Kreinovich 1996; Clausen 1997; Clausen and Perregaard 1997; Kearfott 1997; Adjiman, Androulakis et al. 1998; Adjiman, Androulakis et al. 1998; Clausen and Perregaard 1999; Boyd, Ghosh et al. 2003; Hansen and Walster 2004), and I believe it is also a requirement for any design process to remain rigorous.

While B&B algorithms do use these rigorous bounds, the bounds are computed typically without considering uncertainty. Uncertainty has been incorporated into some B&B algorithms; however, these algorithms are limited to applications where the uncertainty in performance has already been defined for all alternatives (Norkin, Pflug et al. 1996; Montemanni, Gambardella et al. 2004). These applications show that B&B can be applied successfully under uncertainty, but these applications do not consider the many sources of uncertainty encountered in design.

In design, uncertainty arises from many sources (Aughenbaugh 2004). Because of this, the uncertainty from each of these sources must be characterized and incorporated into the resulting bounds in order to remain conservative. To achieve this, I see the following requirements necessary:

1. **Conservatively characterize uncertainty in information used in design.** "Garbage in, Garbage out": a process is only as good as the information one begins with, and the B&B

approach is no different. The uncertain information used has to be characterized conservatively. If this is not done, then the rigor of the process is lost before it even begins.

2. **Apply appropriate models that are conservatively characterized.** In characterizing the performance of a sub region, one must apply models in some form or another. These models, by definition, are abstractions of reality and have some epistemic uncertainty associated with them. This uncertainty must be taken into account in characterizing the performance of a sub region. As a step in accounting for this uncertainty one needs to characterize and validate the model used in computing system performance. Malak suggests a framework for doing so, in which one must incorporate some indication of the uncertainty involved in applying that model, as well as a context for which that model could be applied (Malak Jr. and Paredis 2004). Whether by Malak's framework or another, the models used to characterize performance must themselves be characterized and validated with conservatively characterized uncertainty for a context that includes the context of the design. The model uncertainty can then be appropriately incorporated into the sub region characterization.

3. **Bounds must include all members of that branch.** In branching, the designs are divided into families based on the alternatives or variables being chosen. These families represent all the feasible designs for a particular alternative or variable value. When one bounds the performance of these families, one cannot use one particular instance of the family, instead one must include every member of the family.

4. **Conservatively characterize subset performance.** All of the uncertainty characterized – the uncertainty in information, the model uncertainty, the simulation uncertainty, and

uncertainty about future decisions – needs to be included in the performance characterization of a sub region. Incorporating the uncertainty in this manner is the only way to be conservative in characterizing the performance of a sub region. In addition, this characterization must include all designs considered in that subset

I see these requirements as necessary to result in conservative characterization of the subset performance – my high-level requirement for bounding. While meeting this requirement preserves the rigor of the algorithm, if one only meets these requirements, the efficiency of the algorithm may suffer. To be efficient, I believe one needs to consider the main objectives in this step: to eliminate dominated designs at the least expense. This overall objective can be decomposed into two main objectives:

1. **Avoid being overly-conservative in characterizations.** All the effective conditions are concerned with conservatively characterizing uncertainty; because of this, one may easily fall into the trap of being overly-conservative. This would result in a less-efficient design process just as it results in a less efficient B&B algorithm (Clausen 1997; Hansen and Walster 2004). To see the inefficiency in being overly-conservative, one only needs to look at a simple elimination example. If alternatives *A* and *B* are considered for elimination, and a conservative representation of their performance is given by the intervals $\left[\underline{U}_A, \bar{U}_A\right]$ and $\left[\underline{U}_B, \bar{U}_B\right]$, such that $\underline{U}_A > \bar{U}_B$, then by the conservative bounds one could eliminate alternative *B*, as alternative *B* will always be dominated by *A*. Now, if one was to use overly conservative bounds on just the performance of alternative *B*, such that $\bar{U}_A < \bar{U}_B$. Now one can no longer eliminate alternative *B*, although in reality,

alternative *B* cannot perform that well; the performance of alternative *B* is characterized overly conservative so that the dominance by *A* cannot be established.

Being overly-conservative has a significant negative impact on the efficiency of the B&B approach, but it is not easy to avoid in practice. First, as pointed out in Section 2.1, one often cannot distinguish what is overly-conservative based on the available information. Second, this need to be applied to every step in the process; just as one was conservative with characterizing this information, one needs to avoid being overly conservative.

In being overly conservative, one is not applying all the knowledge and information they posses to determine the bounds on performance. In Chapter 4, I point out another problem that inhibits elimination if one does not include all of their knowledge about uncertainty. I demonstrate how applying this knowledge allows one to eliminate more designs.

2. **Minimize cost of computing characterization.** To compute the performance bounds of a subset, significant resources must be expended. There is a tradeoff in computing these bounds, as recognized in B&B Algorithms by Clausen: "One often experiences a tradeoff between quality and time when dealing with bounding functions: The more time spent on calculating the bound, the better the bound value is." (Clausen 1997). Likewise in design, the designers need to consider the tradeoff between the resources expended in computing bounds and the quality of the bounds.

I believe one should strive to meet these objectives while also holding on to the requirements of this step. In doing so, the process can be rigorous and resource efficient. That rigor is necessary in the bounding step to provide reliable information to use in the elimination and the search strategy. Without reliable information, elimination cannot be rigorous. This is not the only requirement that I see for rigorous elimination; the requirements and objectives that I prescribe for elimination are given next.

### Eliminating in B&B

In the previous steps, the designs were divided into subset and the subset performance was conservatively characterized. Now, the result of those prior steps can be used to eliminate designs and move forward in the design process. In B&B algorithms, this operation often is referred to as pruning and occurs when the subset is infeasibility or dominated (Nemhauser, Rinnooy Kan et al. 1989). Similarly, subsets of designs should be eliminated when they are infeasible or dominated; however, one would not want to eliminate a design that is potentially the most preferred. In the B&B algorithm, this translates into the following criterion: if the performance of subset *A* is $\left[\underline{U}_A, \bar{U}_A\right]$, and the performance of subset *B* is $\left[\underline{U}_B, \bar{U}_B\right]$, and $\underline{U}_A > \bar{U}_B$ then subset *B* should be eliminated. This serves as the basis of my requirements for elimination, which are as follows:

1. **Avoid eliminating feasible designs that are not dominated.** In eliminating, one cannot eliminate designs that are potentially the most preferred, and any design that is not dominated is potentially the most preferred. Unless one can determine conclusively that a design subset is infeasible or is dominated then the design subset cannot be eliminated. This conservative requirement for elimination needs to be upheld, otherwise the search may be unsuccessful at finding the most preferred design. Elimination is investigated in detail in Chapter 4.

2. **Eliminate designs in a consistent manner.** While only dominated designs should be eliminated, one must also ensure that the dominance is determined in a consistent manner. If the criterion used to establish dominance shifts then some of the previously eliminated designs may in fact be the most preferred. To avoid this, one must ensure that designs are evaluated for elimination in a consistent manner.
3. **Integrate with subset characterization (bounding).** In the B&B algorithm, one eliminates subsets based on the bounds of the subsets and the criterion given above. Similarly, one could eliminate designs based on the same criterion. Because of the large uncertainty in design, this criterion may not be effective at eliminating designs, instead I suggest a more effective criterion to eliminated dominated design alternatives that is integrated with the process of bounding and incorporates epistemic uncertainty. This method is derived in Chapter 4.

I believe these requirements ensure that the elimination step does not eliminate any design that is potentially the most preferred, but these requirements alone could result in a uselessly slow design process. To be more efficient, one needs to strive for the following objectives:

1. **Eliminate as much design space as possible.** It is more costly for a designer to consider more designs; therefore the design only wants to consider the designs that could potentially be the most preferred design. The criterion for elimination should eliminate as many designs as possible without eliminating the most preferred design. While this objective is upheld in the criterion used in the B&B algorithm, the objective could be

better achieved by properly integrating the elimination with bounding. This is addressed in Chapter 4.

2. **Simple to Evaluate.** According to Pahl and Beitz, the design process should be easy to learn and teach, as well as be simple to apply (Pahl and Beitz 1996). The more complicated an elimination criterion, the more confusing it is to the designer. If the designer becomes confused or frustrated then one of the benefits of a systematic approach is lost. To minimize confusion, the elimination criterion should be kept as simple as possible. While this objective is important, it must be weighed against the ability to eliminate more designs. This tradeoff is discussed briefly with the elimination method in Chapter 4.

The objectives in elimination can be addressed best by considering how the information available can be used for elimination. In doing so, more elimination is possible without violating the requirements, as is demonstrated in Chapter 4.

### Search Strategy

After the elimination step, the next subset of designs must be chosen to branch. The branch is chosen based on that branch's bounds and the search strategy. This strategy specifies some method to prioritize the subsets to search based on the bounds of the subsets. In the B&B algorithm, this strategy is typically some form of the depth-first, breadth first, or best-first (Nemhauser, Rinnooy Kan et al. 1989; Clausen 1997; Clausen and Perregaard 1999), all of which are reviewed in Section 2.4. While a well-formulated search strategy improves the efficiency of the algorithm, it is not theoretically necessary for the algorithm to converge.

Similarly, the algorithm can converge without a well formulated search strategy, as long as the strategy adheres to the following requirement:

1. **The search method must evaluate all potential designs if necessary.** In order to find the most preferred design, all the designs that could be the most preferred need to be considered. The search strategy must eventually get to all of the remaining subsets of designs. Hopefully, this extensive search will not be necessary, but the search strategy must be formulated to handle it.
2. **The search method must determine when to select a design alternative.** One can eliminate design alternatives to significantly reduce the feasible set of design alternatives. However, using elimination alone, one may not converge on a particular design solution because of the large uncertainty in design. Since elimination may not lead to a particular design solution, one must finish the design process by selecting the design solution from the feasible set. Therefore, the search strategy must include some means of determining when one should select from the feasible set instead of attempting to eliminate further.

Although this requirement may make the search strategy seem simple, that is only because the weight of the strategy is in the objectives. A good search strategy can quickly move through the field to find the most preferred design, whereas a poor one can expend far more resources than necessary. Thus, in formulating a search strategy one should consider the following objectives:

1. **Search should efficiently find the most preferred solution.** In using the B&B approach, one wants to converge to the most preferred solution as quickly as possible and

with the least resources expended. The search method plays an integral part in the efficiency of a B&B algorithm (Nemhauser, Rinnooy Kan et al. 1989; Clausen 1997; Clausen and Perregaard 1999). Therefore, the designers should use a search method that helps eliminate the most designs possible and converge on the most preferred design at the least cost, in the least time. This objective is based on the result of future steps, making it difficult to have the best search strategy and even more difficult to formulate.

2. **Search method for evaluating branches should be clear and consistent.** To avoid confusion or frustration by the designers, the design process should not be more complicated than necessary, including the branching step. The search method should be clear and consistent to avoid designer confusion.

Incorporating these objectives into a search strategy may not be easy, but should result in a significantly more efficient algorithm.

The requirements and objectives presented in this section are based on my own reasoning and my own literature review, therefore they should not be taken as truth. I hope that these requirements can be used to further the research needed to realize this process, or be investigated to produce requirements that can be applied. Either way, these requirements and objectives are presented as the seed of the research process.

### *3.5 Summary of the Branch and Bound Design Process*

I believe a B&B approach to design has some significant advantages, which were elaborated in sections 3.1 and 3.2. Namely, providing designers with a set-based framework to organize the process and specific steps to execute, eliminating much of the confusion in design. In addition, the approach requires the designers to make decisions that reflect the extent of their knowledge and information, thus the designers avoid making decisions that their information does not support and accidentally eliminating the most preferred solution. These advantages cannot be realized without first addressing the challenges in creating the process. These challenges were presented in section 3.3 and highlighted the differences between B&B algorithms and design that would have to be bridged.

The concept of the B&B design process is an approach to design that uses the knowledge from the B&B algorithm to create a rigorous approach to design. The details of this concept are presented in Section 3.4.1. To realize this concept of a B&B design process, I believe the approach must meet my requirements given in subsection 3.4.2. While meeting these requirements would result in a rigorous process, the process would not necessarily be cost effective – for this, one should consider objectives in each step. Specifically, one needs to strive to eliminate as many designs as possible with the least resources; I break this objective into specifics for each step in the process, given in subsection 3.4.2. Both my requirements and objectives are given in Table 7 for each step in the process.

**Table 7: My requirements and objectives in a B&B design approach**

| B&B Design Step | Requirements | Objectives |
|---|---|---|
| Step 1: Branching | 1. The branches must contain the feasible set. | Should create subsets to allow the most elimination possible with the least resources |
| | 2. Each branch must be less inclusive than the feasible set. | |
| | 3. Branching must foster the creative nature of design. | |
| Step 2: Bounding | 1. Conservatively characterize uncertainty in the information used in design | 1. Avoid being overly-conservative in characterizations. |
| | 2. Apply appropriate models that are conservatively characterized. | |

| | | |
|---|---|---|
| | 3. Bounds must include all members of that branch. | 2. Minimize cost of computing characterizations. |
| | 4. Conservatively characterize subset performance. | |
| Step 3: Eliminating | 1. Avoid eliminating feasible designs that are not dominated. | 1. Eliminate as much design space as possible. |
| | 2. Eliminate designs in a consistent manner. | 2. Simple to Evaluate. |
| | 3. Integrate with subset characterization (bounding). | |
| Step 4: Search Strategy | The search method must evaluate all potential designs if necessary | 1. Search should efficiently find the most preferred solution |
| | | 2. Search method for evaluating branches should be clear and consistent |

While the concept of a B&B design approach has significant potential, it is not the main contribution in this thesis. Rather, this work uses the B&B approach to design as the context for looking at the design, giving the perspective needed as the thesis focuses on formalizing the elimination step. This step is the focus for the following reasons:

1. **The other steps in the process are highly influenced by elimination.** Elimination allows the designer to narrow the regions being considered and eventually converge on a design. Without elimination, the process could never converge. Because this step is crucial for the success of the design process, the other steps need to support more elimination. To determine how the other steps should be formulated to achieve this, the elimination step should be formulated first.

2. **Elimination is the most rigorous of the steps.** Dominated designs are eliminated; this is the most rigorous and straight-forward step of the process. Additionally, the previous work in preferences and decision-making can be applied to this problem, thus the basic mathematics needed for formulating the elimination step are already in place. A more rigorous elimination step can be formulated than the other steps, which lack such prior work.
3. **Elimination can be applied independently of B&B.** In a different design approach, one could eliminate dominated designs to converge on the most preferred design. In such a process, a properly formulated elimination step is the most important component. The elimination method has value beyond the B&B process.

For these reasons, the contribution in this thesis is focused on creating a rigorous method of elimination integrated with bounding. This is undertaken in the next Chapter. In this chapter, my concept of a Branch and Bound approach to set-based design was presented, and my requirements and objectives for the approach were stated. This introduction of the B&B approach to design has given the context for looking at the design process. Attention is focused on elimination, in the next chapter, as shown in Figure 22.

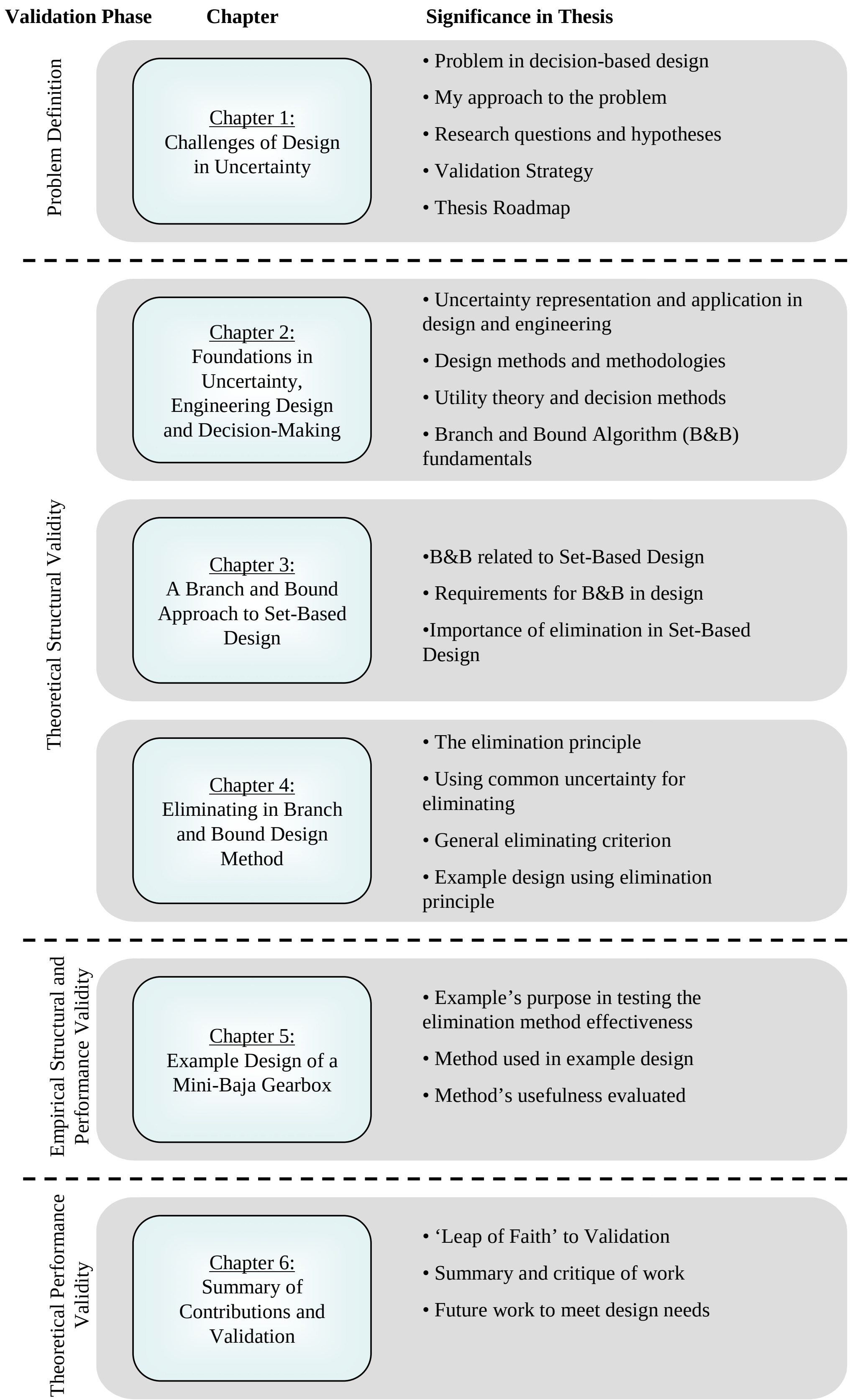



***Figure 22: Thesis Roadmap***

# CHAPTER 4

# ELIMINATION UNDER INTERVAL-BASED UNCERTAINTY

In the previous chapter, the merits offered by applying Branch and Bound (B&B) in design were shown, but as was pointed out, to obtain these benefits, one must efficiently eliminate design alternatives. Effective elimination is important in any set-based approach to design and motivates the investigation of elimination. A principle for elimination under interval-based uncertainty is developed in this chapter.

The purpose of this chapter is to develop an elimination principle under interval-based uncertainty that is theoretically sound and useful. In developing this principle, the hypothesis from Chapter 1 is addressed. The research question and hypothesis are reiterated in Section 4.1. To address the issues in this hypothesis, Section 4.2 is dedicated to developing and supporting the elimination principle, while Section 4.3 shows how to apply knowledge about uncertainty in elimination. These results are combined in Section 4.4 to complete the picture, as the Q1 and H1 are revisited. Section 4.5 summarizes the results and ties Chapter 4 into the validation picture and thesis roadmap.

## *4.1 Basis for Eliminating Designs*

The design process is based on eliminating many potential designs to result in a single design. Since elimination is the objective in the design process, efficient and effective design requires sound elimination. To this end, this thesis is focused on elimination in design, and elimination is the focus of the primary research question:

**Question:** *Under conditions of interval-based uncertainty, how should one eliminate designs?*

In design, one should eliminate all but the best performing design. To determine the best performing design, one compares the designs based on their performance attributes. These attributes must be weighed against each other to determine the design's overall performance. Many possible schemes exist to weigh the importance of the design's attributes and produce a measure of overall performance; however the designer should weigh these attributes consistent with their preferences. Evaluating designs consistent with one's preferences is rational behavior and at the heart of Utility Theory (von Neumann and Morgenstern 1944; Luce and Raiffa 1957; Keeney 1974; Keeney and Raiffa 1993; Fernandez, Seepersad et al. 2001).

According to Utility Theory, a person seeks to maximize their utility (von Neumann and Morgenstern 1944; Luce and Raiffa 1957; Keeney 1974; Keeney and Raiffa 1993; Fernandez, Seepersad et al. 2001), and according to von Neumann and Morgenstern, a numerical value of utility can be assessed based on rigorous mathematical rules (von Neumann and Morgenstern 1944). When this is applied to the design process, the designer maximizes his utility derived from the designed artifact; when judged from the designer's perspective, the higher the utility from the designed artifact the better the performance of that design. For two designs, *A* and *B*, if $U(A) > U(B)$ then *A* performs better than, or dominates, *B*; this dominance can be expressed, as follows: $A \succ B$. In this situation where $A \succ B$, *B* cannot possibly perform as well as *A*, therefore *B* should be eliminated; this elimination is rational.

In rational elimination, if one is not prudent in applying the knowledge and information then one may not be able to eliminate. Therefore one should not only be rational in eliminating, but one should also apply all the knowledge and information available towards eliminating. This leads to the primary hypothesis:

> **Hypothesis:** *One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design and by accounting for shared uncertainty.*

Hypothesis 1 has three specific aspects to be investigated that relate to how one eliminates rationally and how one applies knowledge about shared uncertainty toward eliminating. These two aspects are addressed in two sections. Section 4.2 covers rational elimination under interval-based uncertainty, while Section 4.3 covers how to apply knowledge about shared uncertainty toward eliminating, and Section 4.4 explains why one should compare the performance to a detailed, specific reference design. In Section 4.5, I tie everything together.

## *4.2 The Principle for Elimination under Interval-Based Uncertainty*

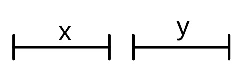
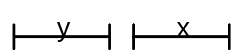
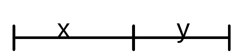
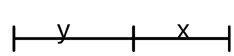
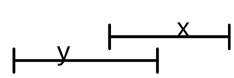
As explained in Chapter 2, the focus of this work is on uncertainty that is represented by an interval. When propagated through performance models, interval-based uncertainty results in an interval on possible design performance. In this work, I assume that these interval bounds on performance can be computed. Thus, rational elimination needs to be applied to these performance intervals.

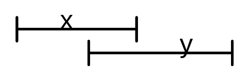
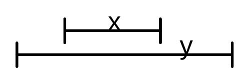
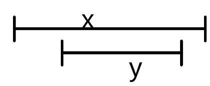
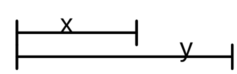
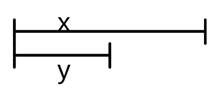
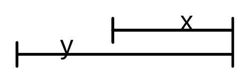
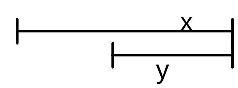
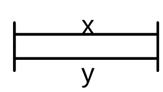
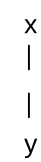
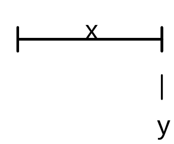
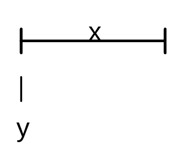
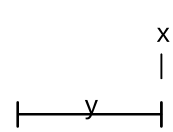
Intervals complicate elimination. In an interval, all outcomes in that interval are possible, and none is known to be more likely than any of the others, therefore one must consider all the possible outcomes when eliminating. In eliminating rationally, one eliminates a design if that design is dominated, that is, guaranteed to be less prepared than at least one other design. Thus, in order to eliminate a design based on an interval of performance, the design must be dominated for all possible outcomes, as stated:

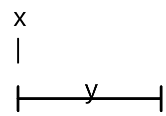
> **Elimination Principle** One should eliminate a design alternative if, and only if, that design alternative is dominated by at least one other design alternative throughout all uncertainty conditions.

As pointed out in the elimination principle under interval-based uncertainty, dominance has to be established under all uncertain conditions. Since interval-based uncertainty results in performance characteristics in terms of intervals, one must establish dominance by comparing intervals. However, comparing intervals is not as simple as comparing scalars. For comparing intervals, there are 18 different basic relations, as pointed out by (Hayes 2003). These relations are given in Table 8.

**Table 8: Interval Relations (Hayes 2003)**

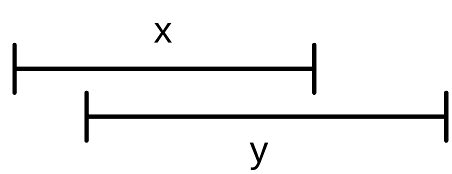
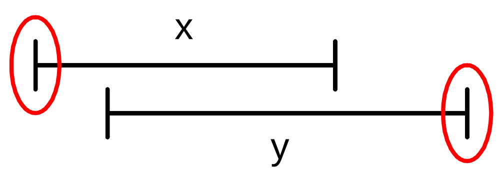
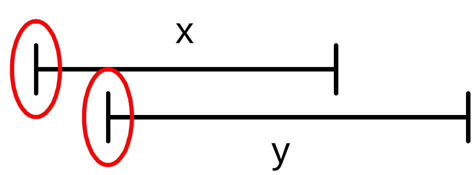
| Relation | Constraint | Meaning |
|---|---|---|
| X <<<< Y | $\overline{X} < \underline{Y}$ |   |
| X >>>> Y | $\overline{Y} < \underline{X}$ |   |
| X <<<= Y | $\overline{X} = \underline{Y}$ |   |
| X =>>> Y | $\overline{Y} = \underline{X}$ |   |
| X <>>> Y | $\underline{Y} < \underline{X}$ and $\underline{X} < \overline{Y} < \overline{X}$ |   |

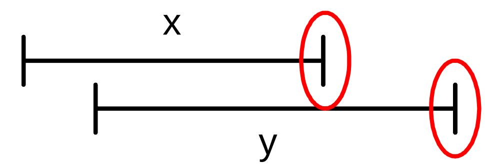
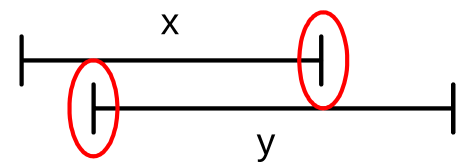
| | | |
|---|---|---|
| X <<<> Y | $\underline{X} < \underline{Y}$ and $\underline{Y} < \overline{X} < \overline{Y}$ |  |
| X <><> Y | $\underline{Y} < \underline{X}$ and $\overline{X} < \overline{Y}$ |  |
| X <<>> Y | $\underline{X} < \underline{Y}$ and $\overline{Y} < \overline{X}$ |  |
| X <=<> Y | $\underline{X} = \underline{Y}$ and $\overline{X} < \overline{Y}$ |  |
| X <=>> Y | $\underline{Y} = \underline{X}$ and $\overline{Y} < \overline{X}$ |  |
| X <>=> Y | $\underline{Y} < \underline{X}$ and $\overline{X} = \overline{Y}$ |  |
| X <<=> Y | $\underline{X} < \underline{Y}$ and $\overline{Y} = \overline{X}$ |  |
| X <= => Y | $\underline{Y} = \underline{X}$ and $\overline{Y} = \overline{X}$ |  |
| X = = = =Y | $\underline{X} = \overline{X} = \underline{Y} = \overline{Y}$ |  |
| X < < = =Y | $\overline{X} = \underline{Y} = \overline{Y}$ |  |
| X = = > >Y | $\underline{X} = \underline{Y} = \overline{Y}$ |  |
| X = > = >Y | $\underline{X} = \overline{X} = \overline{Y}$ |  |

| | | |
|---|---|---|
| X < = < =Y | $\underline{X} = \overline{X} = \underline{Y}$ |  |

As seen in Table 8, each interval relations consist of 4 scalar relations. These 4 scalar relations compare the same bounds on the two intervals in the relation. For example, the first scalar relation in the interval relation compares the lower bound of the first interval to the upper bound of the last interval. The other relations follow similar rules, as diagrammed in Table 9. With this understanding of the interval relations, attention is turned back eliminating based on these interval relations.

**Table 9: Meaning of Interval Relations**

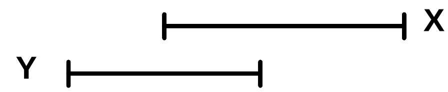
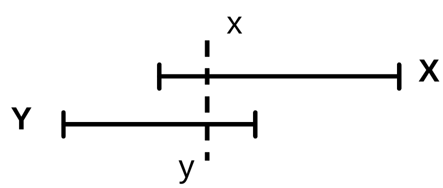
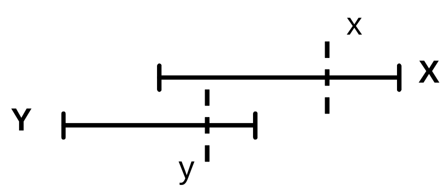
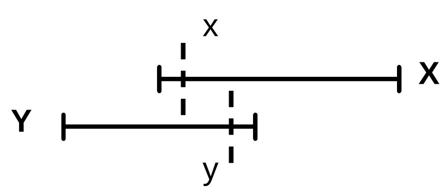
| **Interval Relation** | **Constraint** | **Meaning** |
|---|---|---|
| X <<<> Y | $\underline{X} < \underline{Y}$ and $\underline{Y} < \underline{X} < \overline{Y}$ |  |
| **Component in Interval Relation** | **Scalar Relation** | **Meaning of Scalar Relation** |
| X **<**<<> Y | $\underline{X} < \overline{Y}$ |  |
| X <**<**<> Y | $\underline{X} < \underline{Y}$ |  |

| | | |
|---|---|---|
| X <<<> Y | $\bar{X} < \bar{Y}$ |  |
| X <<<> Y | $\bar{X} < \underline{Y}$ |  |

Based on the interval relations given in Table 8, one must establish dominance for all possible outcomes in both intervals. However, for most of these interval relations one cannot determine dominance for the outcomes. Specifically, for all intervals that overlap, one is not able to determine a relation between two instances in those intervals. This concept is shown for the X <>>> $Y$ relation and their instances, $x \in X$ and $y \in Y$, in Table 10. At first one may think $x > y$, but this would be a misinterpretation of the relation. The relation X <>>> $Y$ means that *x* has the *possibility* of being greater than *y*, but will not *necessarily* be greater than *y*. Nor is *x* known to be more likely to be greater than *y*. In other words, $x > y$, $x < y$, and $x = y$ are all possible outcomes, and one does not know which outcome is more likely. This is a crucial aspect in comparing instances based on their intervals: if the intervals overlap then no conclusions can be drawn about the relation of their instances.

**Table 10: Relation of Instance of Intervals that Overlap**

| Interval Relation | Constraint | Meaning |
|---|---|---|

<table>
<tr><td>$X <>>> Y$</td><td>$\underline{Y} < \underline{X}$ and $\underline{X} < \overline{Y} < \overline{X}$</td><td></td></tr>
<tr><td colspan="3">Let $x \in X$ and $y \in Y$</td></tr>
<tr><td>Possible Instance Relations</td><td colspan="2">Meaning of Instance Relations</td></tr>
<tr><td>$x = y$ where</td><td colspan="2"></td></tr>
<tr><td>$x > y$</td><td colspan="2"></td></tr>
<tr><td>$x < y$</td><td colspan="2"></td></tr>
</table>

Since most interval relations involve overlap, only for the $X >>>> Y$ and $X <<<< Y$ interval relations can one determine a relation between the instances in the intervals. In other words, if $X >>>> Y$ and $x \in X, y \in Y$ then $x > y$. For the $X >>>> Y$ and $X <<<< Y$ interval relations, and these only, one can eliminate. This criterion is stated as follows:

**Elimination Criterion**

Consider the choice between two alternatives, $A$ and $B$ with $U(A) \in [\underline{U}(A), \bar{U}(A)]$ and $U(B) \in [\underline{U}(B), \bar{U}(B)]$.

One should eliminate alternative B if, and only if:

$$[\underline{U}(A), \bar{U}(A)] >>>> [\underline{U}(B), \bar{U}(B)]$$

or:

$$\underline{U}(A) > \bar{U}(B)$$

With this decision criterion there are situations in which one cannot eliminate a design, as establishing dominance between alternatives is not always possible, therefore the results of applying this criterion is the solution set in which that distinction could not be made. The concept of using non-dominated sets is not new. In fact, von Neumann and Morgenstern proposed this idea in the development of utility theory for a solution set, S: "The elements of S are precisely those elements which are undominated by elements of S." (p. 40 in (von Neumann and Morgenstern 1944))

In situations that von Neumann and Morgenstern encountered, the uncertainty was aleatory in nature. With aleatory uncertainty, one applies the expected-utility theorem, as is presented in (von Neumann and Morgenstern 1944; Luce and Raiffa 1957; Keeney 1974; Keeney and Raiffa 1993; Triantaphyllou 2000; Stirling 2003), as the decision criterion. With this criterion, one chooses the alternative with the highest expected utility: if $E(U(A)) > E(U(B))$ then $A \succ B$. By using this criterion, one is selecting the alternative that is most-likely to perform the best. This is stated: $E(U(A)) > E(U(B)) \Rightarrow P(U(A) > U(B)) > P(U(A) < U(B))$. This criterion has the advantage of eliminating all but a single alternative unless the unlikely

$E\left(U\left(A\right)\right)=E\left(U\left(B\right)\right)$ situation occurs. Although this decision criterion is consistent with the probabilistic information, it does not apply when the utility is an interval.

With intervals, there is no information about the probability of outcomes; therefore one cannot determine which alternative is more likely to perform better. In these situations, a single solution cannot always be reached. This inability to eliminate under interval-based uncertainty is contrasted with probability-based uncertainty in *Figure 23*. With intervals, the results of the alternatives are not sufficiently known to establish that one is superior, or even more likely to be superior. Since one does not know which alternative is superior, eliminating in this situation is not possible. When one cannot decide on a unique design, what should one do?

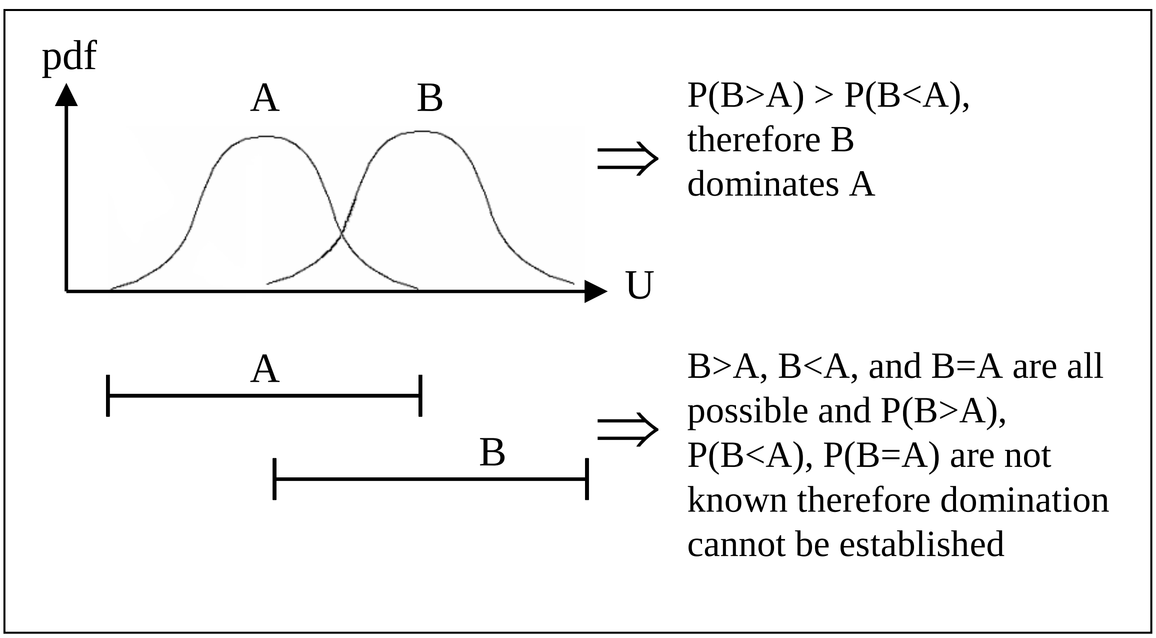



*Figure 23: Eliminating with probability-based uncertainty is eased by information about the distribution within the unknown; interval-based uncertainty lacks this information and decisiveness*

One possible approach is to delay further elimination to future design decisions. Delaying elimination is not a problem – one does not have to eliminate all but one design in a single decision. A single decision in the design process will not unambiguously finalize a design; therefore a single design does not have to result from a decision. The decision can be delayed until more information is available. As more information becomes available in the design process, the designers can make more informed decisions.

An example of such an approach can be found at Toyota. Toyota engineers delay design decisions until they can be certain of their decision. These decisions are being delayed because the engineers cannot determine a single, dominant alternative; this is consistent and supportive of the elimination principle. When decisions are delayed, tests or other investigations are performed to reduce uncertainty, allowing the engineers to make more informed decisions, as is commended in chapter 2 (Ward, Liker et al. 1995; Sobek and Ward 1996; Sobek II 2004). As pointed out in chapter 2, Toyota has shown how delaying design decisions can have a positive impact on the resulting design.

In many circumstances, one has difficulty eliminating alternatives, thus the objective in elimination should be to reduce the design space considered as much as possible. In the multiple-alternative and continuous cases, there is significant design space that can be reduced in elimination without making an unambiguous decisions. Eliminating in the multiple-alternative case is similar to two-alternative case, but the elimination criterion must be extended to apply to all the alternatives. To include all the alternatives in the decision criterion, one does not need to compare all the alternatives to each other; instead an alternative can be chosen as reference by which all designs are compared for elimination.

Elimination can only occur if the lower bound of one design is larger than the upper bound on the design being eliminated; therefore the design with the maximum lower-bound of performance from all the alternatives can be used to compare against all other designs for elimination. No other comparisons could result in additional designs being eliminated. This concept is shown in Figure 24. In this figure, Alternative 2, with the maximum lower bound eliminates alternative 1, 4, and 5. No additional elimination is possible if any other reference is used, and in fact, if any other reference design is used then alternative 4 could not be eliminated. This demonstrates how the alternative with the maximum lower bound is the only reference design needed for elimination.

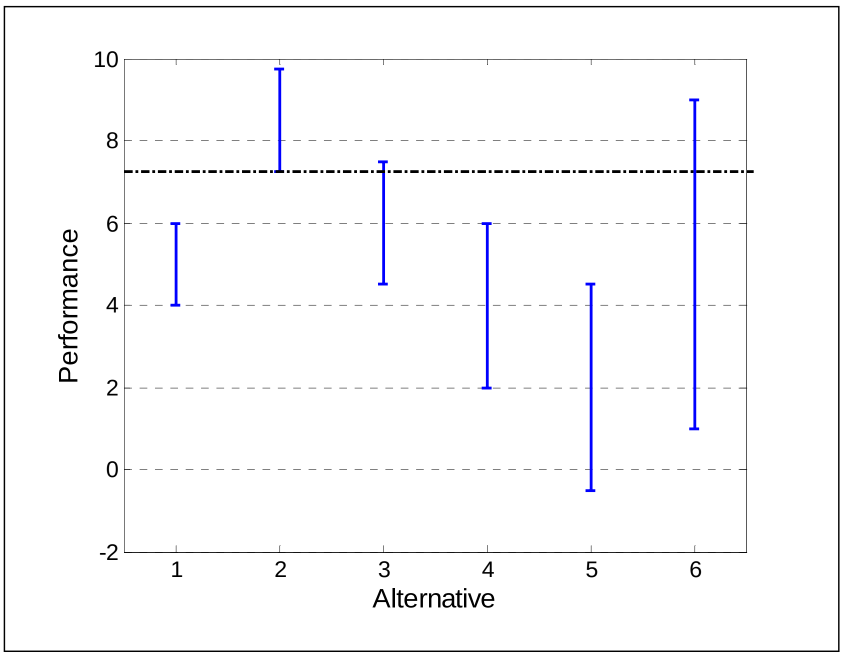



***Figure 24: The alternative with the maximum lower bound can be used as a reference to eliminate.***

The concept shown in Figure 24 not only applies to the multiple-alternative case but also can be used in the continuous case. For the continuous case, the maximum lower-bound is found not from a specific alternative but instead from a continuous region. The relation between the continuous case and the multiple-alternative case is shown in Figure 25. This figure shows how the continuous case can be divided into multiple regions. The same elimination criterion for the multiple-alternative case can then be applied. This figure also shows that it is most beneficial to choose the maximum lower-bound should to eliminate a region of the design space from consideration. Since elimination can only be justified if the lower bound of the reference design is larger than the upper bound on the design being eliminated, any reference design other than the one with the maximum lower bound would result in fewer eliminations. Thus, for both the multiple-alternative and continuous cases, one should use the maximum lower-bound found throughout the possible designs as the reference. Based on this reference, the general criterion for eliminating is presented in Figure 26.

The principle and criterions given allows one to eliminate the dominated alternatives or regions; however, how the regions or alternatives should be created for elimination has not been addressed. This is a branching issue that is left for future work and is enumerated in the Chapter 6. Attention is turned toward demonstrating this elimination principle in a simple engineering problem.

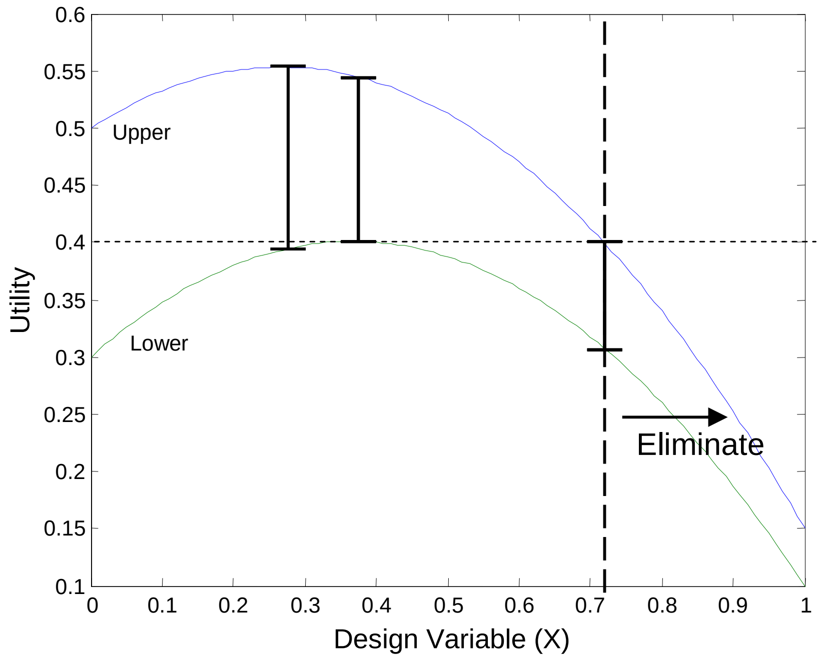



***Figure 25: The maximum lower bound can be used as a reference to eliminate.***

***Elimination Criterion***

*Consider design space D. Let* $A_i \in D$ *with* $U\left(A_i\right) \in \left[\underline{U}\left(A_i\right), \bar{U}\left(A_i\right)\right]$.

*Eliminate* $A_j$ *if and only if* $A_j$ *is dominated by at least one other design:*

$$\exists A_i \in D : A_i \succ A_j$$

*or:*

$$\bar{U}\left(A_j\right) < \max_{A_i \in D} \underline{U}\left(A_i\right)$$

***Applying the Criterion***

*All alternatives in the set D should be eliminated in accordance with the elimination criterion such that the remaining alternatives form a non-dominated solution set* $S \subset D$ *with:*

$$\forall A_i, A_j \in S : A_i \nsucc A_j$$

**Figure 26: The elimination criterion for interval-based uncertainty.**

**Brief Example – Beam design**

A practical problem is needed to demonstrate the given elimination principles. For this purpose the selection of a beam was chosen for its simplicity, so the method is not lost in the complexity of the problem. The problem considered involves the selection of a beam to support a load, as diagrammed in Figure 27.

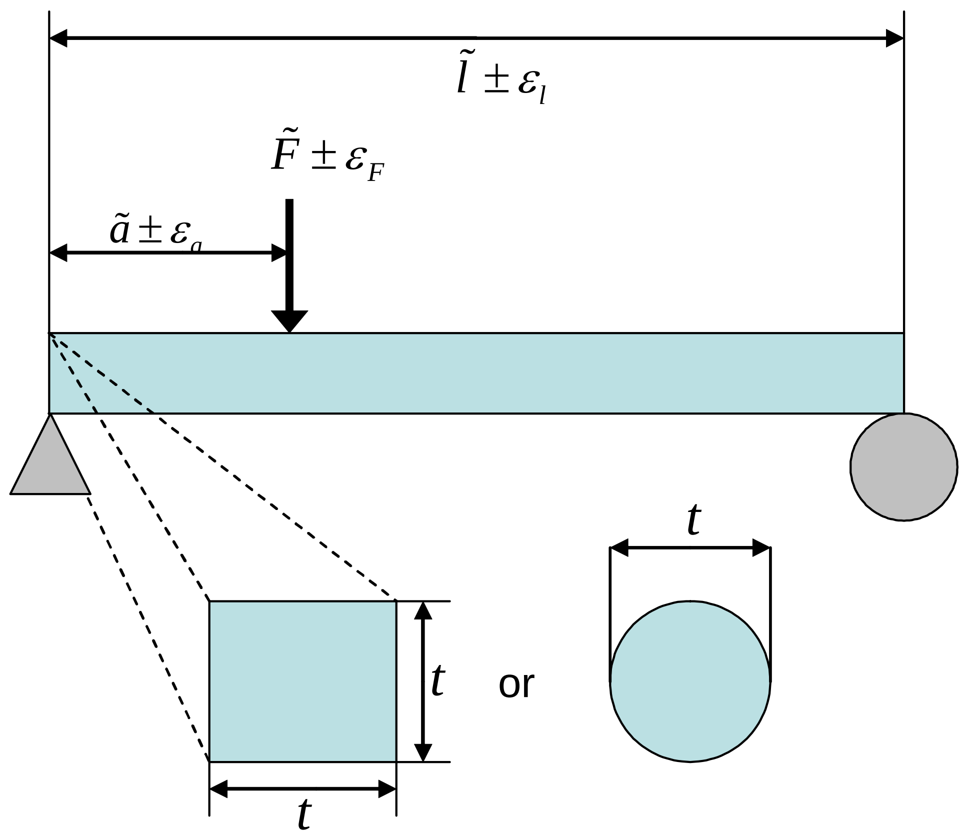



***Figure 27: Diagram of the beam design example***

In this design example, a load of uncertain magnitude is applied to a simple beam (Riley, Sturges et al. 1995). The beam material and beam shape are chosen to minimize the cost as well as maximize the safety factor. This design problem is summarized in the problem formulation in *Figure 28*.

In *Figure 28*, the goal in the design is presented as maximizing the utility. The utility function trades off lower cost versus a larger safety factor in an exponential manner that was arbitrarily chosen by the author. The utility function with respect to cost and safety factor is shown in Figure 29. The utility from each of these attributes are weighed equally in the overall utility function.

To model the cost, the relative costs of the materials were extracted from Ashby (Ashby 1999); these costs are relative to that of mild steel. The relative costs are then multiplied by the mass of the respective beams to determine the relative cost of the beams.

***Maximize***

*Utility*

$U = \frac{1}{2}(1 - e^{-N_f} + e^{-C})$ , *where* $N_f$ *is the safety factor and* C *is relative cost*

***Select***

*Beam Material*

Aluminum,Steel,Wood,Titanium

*Beam Thickness (t)*

$t = [0.1,\ 0.25]\,m$

*Beam Shape*

Solid Square: $t \times t$

Circular: diameter = t

***Where***

*Cost is* $C = C_{\text{mat}}\rho_{\text{mat}}V$

$\rho_{\text{mat}}$ *= the material density*

$V$ *= the volume of the beam*

$C_{\text{mat}}$ *= relative material cost for each material extracted from Ashby (Ashby 1999).*

*Safety factor is* $N_f = \frac{S_y}{\sigma}$

$S_y$ *is the yield strength*

$\sigma$ *is the bending stress, calculated from:*

$$\sigma = \frac{Fa(L - a)\left(t/2\right)}{2IL}$$

$F = [15000, 30000]\,N$ *is the force*

$a = [0.5,\ 3.5]\,m$ *is the position of the force*

$L = 10m$ *is the beam length*

I *is the moment of inertia of the beam cross section, given by:*

$I = \frac{\pi\left(t/2\right)^4}{4}$ *for circular cross sections*

$I = \frac{t^4}{12}$ *for square cross sections*

*Figure 28: Formulation of the beam design example*

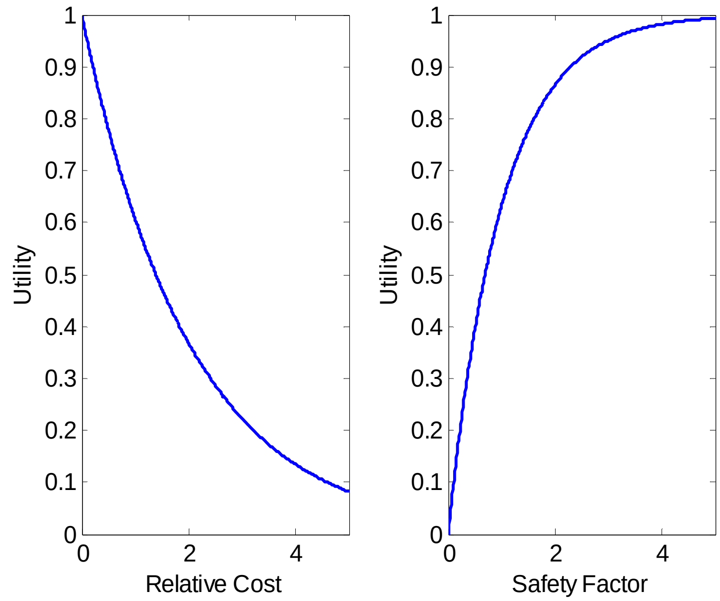



***Figure 29: Attribute Utility Functions for Beam Example***

To compute the safety factor the beam dimensions were used to determine the moment of inertia of the beam cross section (Riley, Sturges et al. 1995):

$$\text{Circular: } I = \frac{\pi \left( t/2 \right)^4}{4}$$

where $t$ is the diameter of the beam cross section and $I$ is the moment of inertia.

$$\text{Square: } I = \frac{t^4}{12}$$

where $t$ is the edge length of the cross section and $I$ is the moment of inertia. Using these moments of inertia for the respective beams, the normal stress is calculated (Riley, Sturges et al. 1995):

$$\sigma = \frac{Fa\left(L-a\right)\left(t/2\right)}{2IL}$$

where $F$ is the force applied to the beam, $L$ is the length of the beam, $a$ is the distance from the end of the beam to where the force is applied, and $\sigma$ is the maximum normal stress in the beam. This stress is used with the respective material strength to calculate the safety factor for the beam:

$$\text{Safety Factor} = \frac{S}{\sigma}$$

where S is the strength of the material and $\sigma$ is the stress for that shape. With these models of system performance in place, and the uncertainty expressed, the Branch and Bound paradigm can be applied.

In applying the Branch and Bound, the first step is to branch the design space. For this, the beam material is chosen based on the experience of the author. This branching is shown with respect to the process and the design space in Figure 30. In the left part of this figure, the branch to the different regions is shown while the branched design space is shown on the right part of the figure. In the next step, bounds are determined for each of the branched regions.

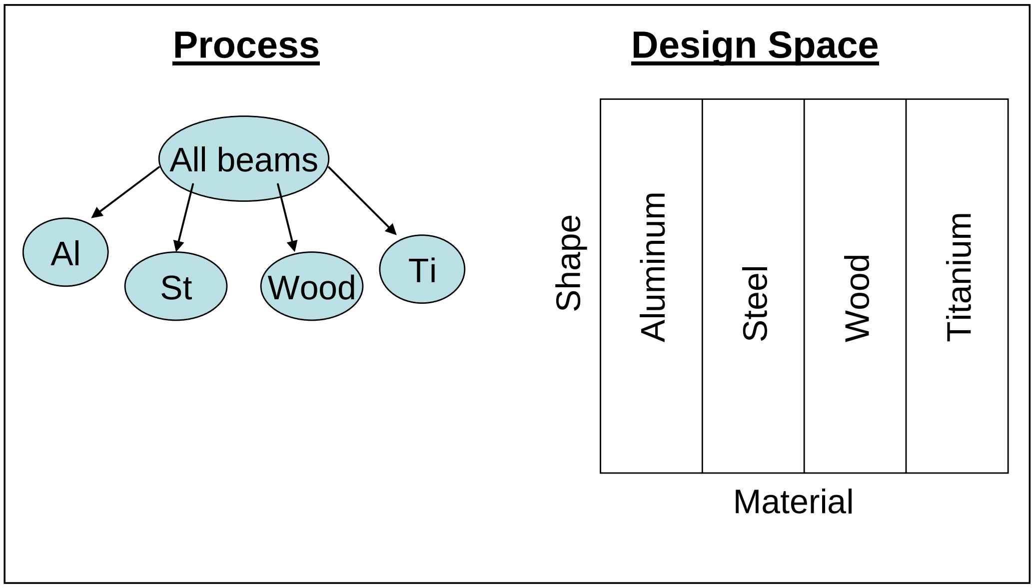



***Figure 30: Branching in the design process and the design space***

Branch and Bound, as applied in optimization and search algorithms, typically uses a deterministic objective function, as pointed out in Chapter 2. However, in this application and in design in general, the objective function is not deterministic. Rather, the uncertainty in the environmental parameters causes uncertainty in the objective function results. This difference does not cause any problems with the algorithm, but changes how bounds must be calculated for a region, as pointed out in Chapter 3.

To calculate the bounds on a region of design space, one must not only take into account the different designs within a region, but one also must consider the uncertainty. Thus, the bounds on a region must be computed for all the possible designs in that region, under all possible conditions of uncertainty. A search this large would be difficult if it were not for some assumptions that can be made about the space being bounded.

Fortunately, in many situations, the utility function being bounded is monotonic with respect to the design variables and uncertainty. A monotonic function has a first derivative that does not change sign, which guarantees that the bounds of the function are at the boundary of the region being searched. This concept is shown by comparing monotonic functions against a non-monotonic function in Figure 4.7. The functions on the left of this figure are monotonic, thus the bounds occur at the boundary of the region. However, the function in the figure on the right is non-monotonic, therefore the bounds do not necessarily occur at the boundary. In this case neither bound is at the boundary of the variable range.

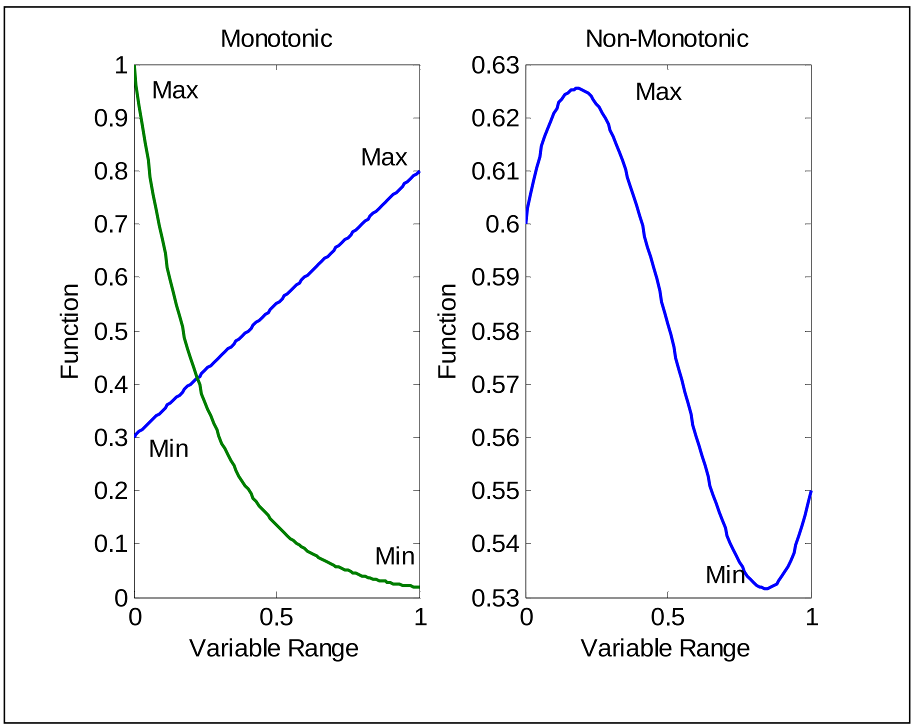



***Figure.31: Monotonic vs. Non-Monotonic Functions for finding maximum and minimum***

In the beam example, the utility function is monotonic with respect to the uncertain variables, therefore the bounds on the performance occur at the bounds of the uncertainty. An example of this is in Figure 32, where the square aluminum beam is shown to be monotonic with respect to both the force, *F*, and distance along the beam, *a*. Similar to the square aluminum beam, the utility of the other beams are also monotonic with respect to the uncertain parameters. For all of these beams, the maximum performance occurs at the minimum possible force and minimum distance along the beam; the lower bound on performance occurs at *Force* = 15kN and *a* = 0.5m. Conversely, the conditions for minimum performance occur at the maximum force and maximum distance along the beam: *Force* = 30kN and *a* = 1.5m. These conditions are used in computing the utility bounds for each material, but as pointed out above variation in performance throughout the branched regions still needs to be taken into account. To do so, the minimum and maximum conditions are used to calculate performance for each shape. From these results the bounds are determined for each material. The utility bounds with respect to each material are shown in Figure 33.

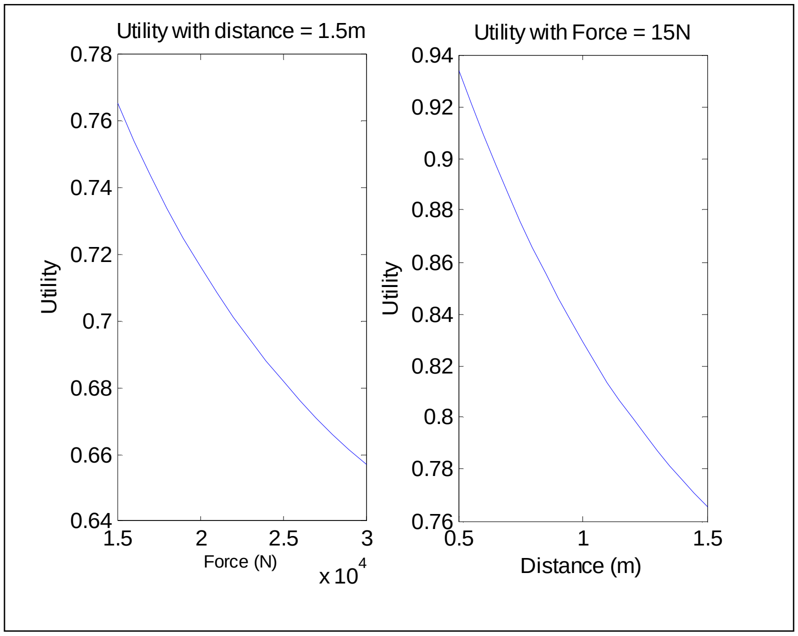



***Figure 32: Monotonic Utility Function for the Beam with Respect to Uncertain Conditions***

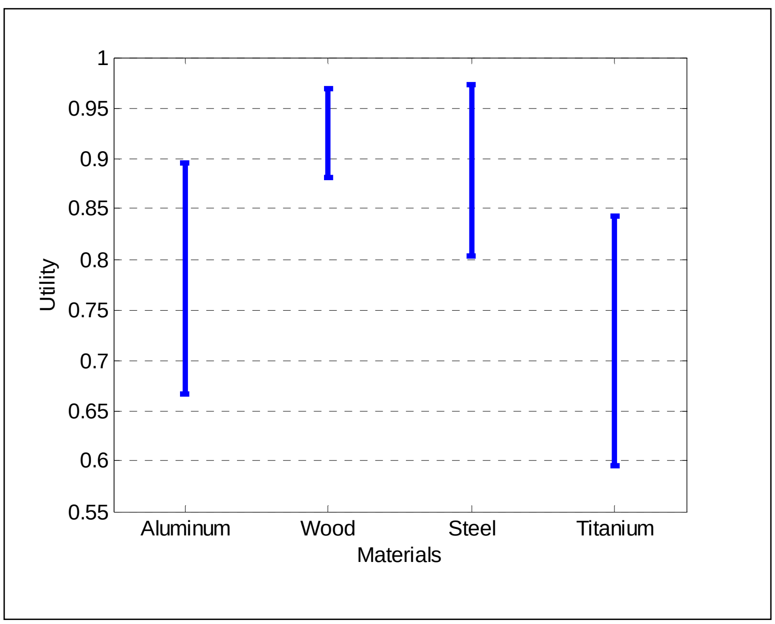



***Figure 33: Bounds on Beam Utility with respect to Materials***

Based on the bounds in Figure 33, one can eliminate the titanium beam from consideration, since under no conditions in the example would titanium perform as well as wood. The other materials must remain in the set, as one cannot be sure that any of these alternatives are dominated or dominate. The results of this elimination are seen in the process and design space shown in Figure 34. As shown in this figure, Titanium is no longer in consideration.

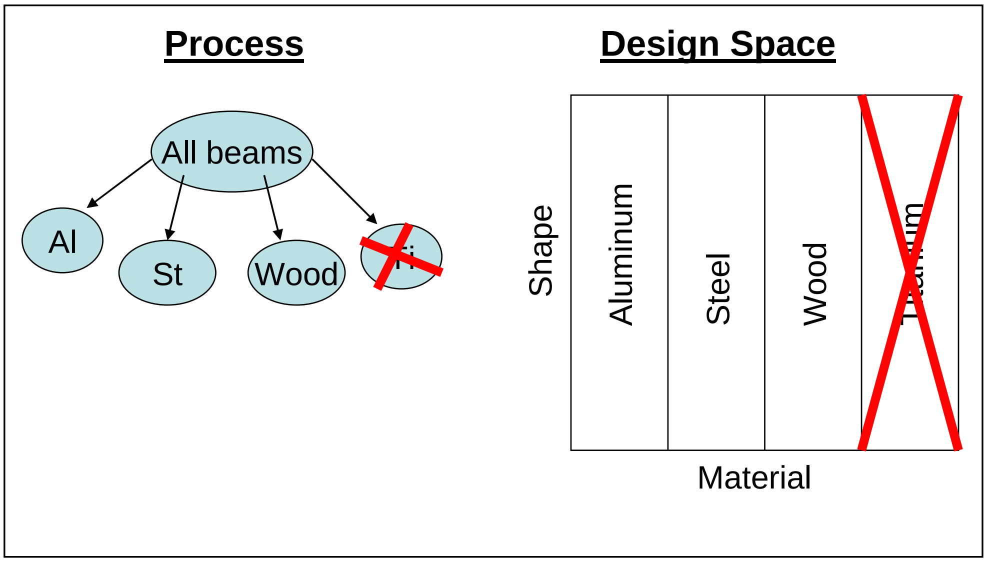



***Figure 34: The design process and design space after the first branch and elimination***

The next step in the process is to branch again. At this time there are multiple branching possibilities in the B&B paradigm; one could select to branch on any variety of regions for a variety of different reasons. These branching possibilities are left as future work, while the author has chosen in this example to branch in each region with respect to beam shape for each material; this next step is shown in Figure 35.

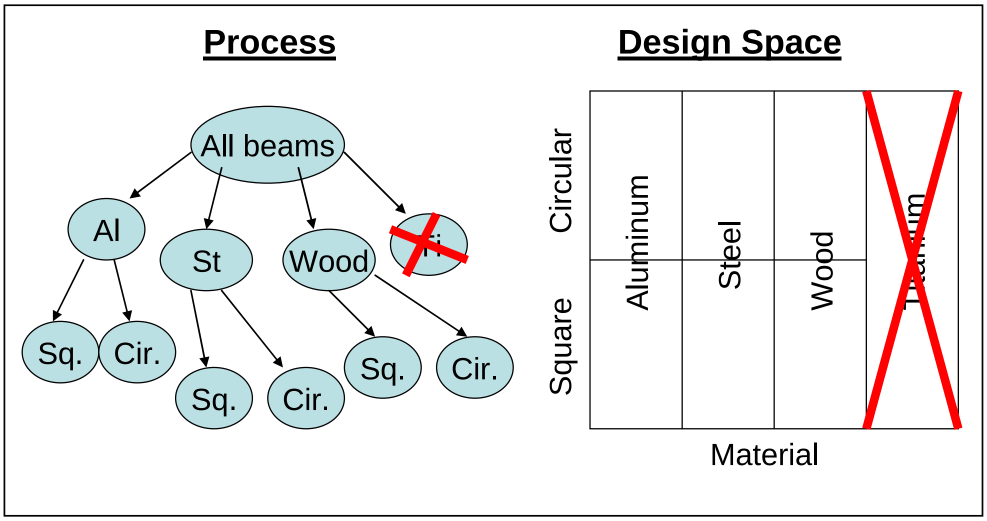



***Figure 35: The design process and design space after the second branching***

For each of these regions the bounds are computed in the same manner as previously, using the same conditions as before to compute the maximum and minimum performance. These performance bounds are shown in Figure 36.

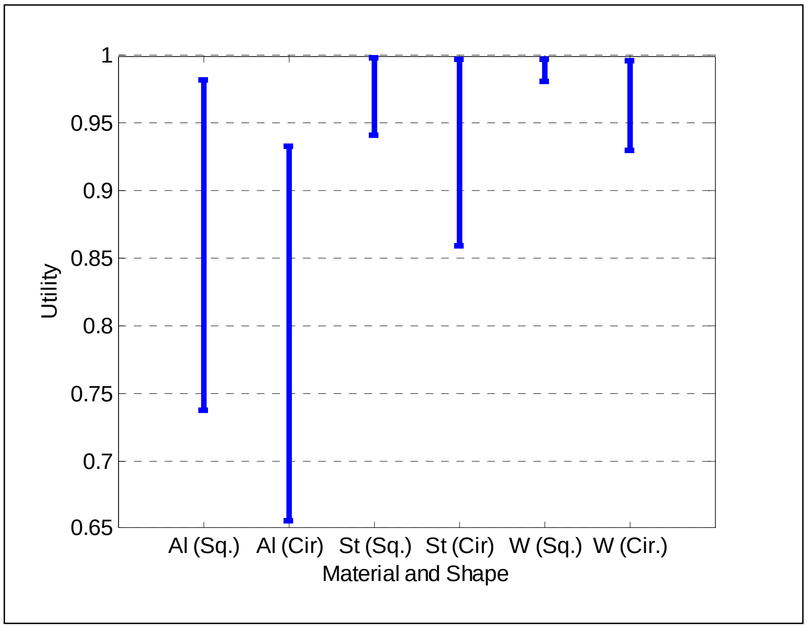



***Figure 36: Bounds on Utility with respect to Material and Shape***

Based on the bounds displayed in Figure 36, one eliminates the aluminum circular beam; this elimination is shown with respect to the design space and branches in Figure 37. Although there still are a significant number of designs that remain, no further elimination is possible. This may seem like one of the shortcomings in applying this decision criterion under conditions of interval-based uncertainty; however, based on the information available and the knowledge used so far, further eliminating would risk eliminating the best solution. This would be unacceptable. Rather than taking a chance and making an unsupported decision, attention is turned to the question: Is there additional knowledge available that allows us to eliminate additional design alternatives?

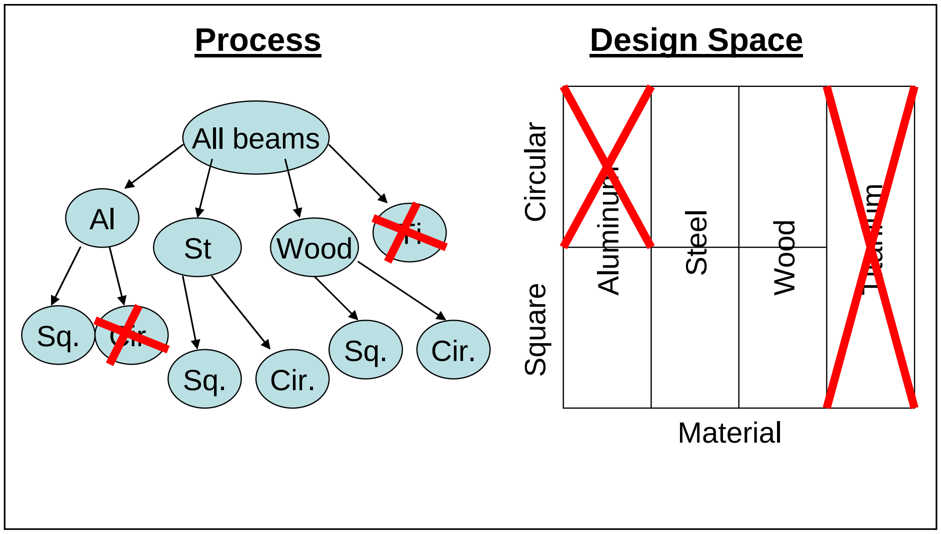



***Figure 37: The design process and design space after the second branching and elimination***

## *4.3 Improving Elimination: Additional Knowledge about Uncertainty*

As pointed out in the previous section, uncertainty in design appears to limit the opportunity for elimination severely. However, there is additional knowledge that can be applied allowing further elimination without the risk of eliminating the best solution. In this section, it is demonstrated that taking additional knowledge about shared uncertainty into account often allows significant additional elimination.

### 4.3.1 Applying Knowledge about Shared Uncertainty

Often in design, there are environmental conditions that, although uncertain, are the same for all the designs being compared; these environmental conditions are shared uncertainty. Shared uncertainty is any uncertain parameter that has the same value for all the designs being compared. An example of this is ambient temperature; ambient temperature, although uncertain, is the same for any design considered; however, commonality like this is often ignored; in these cases, different values for the shared parameter are used, and in these cases, designs are not compared under the same uncertain conditions. Ignoring shared uncertainty in this way unnecessarily limits elimination.

When shared uncertainty is ignored, the reference design is compared under the worst uncertainty conditions while the design considered for elimination is compared under the best uncertain conditions. Although this is conservative, it is makes elimination more difficult. However, if shared uncertainty is considered, the designs are compared under the same conditions, and more elimination is possible. The details of how this is possible are explained mathematically.

Consider the equation $y = 2x - x$ where $x \in [-1,1]$. If one computes bounds on *y* without considering *x* to be the same instance, then the result is as follows:

$$
\begin{aligned}
y &= 2 * [-1,1] - [-1,1] \\
y &= [-2,2] - [-1,1] \\
y &= [-3,3]
\end{aligned}
$$

These bounds on *y* are overly conservative because the uncertain variable, *x*, has been treated as two separate instances. This is equivalent to $y = 2x - z$, where $x \in [-1,1]$ and $z \in [-1,1]$. In Interval Analysis, this problem is referred to as the dependence problem (Kearfott and Kreinovich 1996; Hansen and Walster 2004). To avoid this problem, eliminate the repeated variable *x* to calculate the bounds on *y*, as follows:

$$
\begin{aligned}
y &= 2x - x = x \\
y &= [-1,1]
\end{aligned}
$$

This results in appropriate bounds on *y* by using the same instance of *x* in the equation. To avoid being overly conservative, one must consider this dependence problem when computing with intervals. This idea is applied to calculating system performance by using the same instance of an uncertain parameter when computing with shared uncertainty.

The same idea has been applied before in probabilistic simulation in the form of common random numbers. Law and Kelton  give the crux of common random numbers: “The basic idea is that we should compare the alternative configurations ‘under similar experimental conditions’ so that we can be more confident that any observed differences in performance are due to differences in the system configurations rather than to fluctuations of the ‘experimental conditions’.” (Law and Kelton 2000) The basic idea is the same as that for shared uncertainty; however common random numbers are specific to probability-based uncertainty, whereas shared uncertainty is applied to interval-based uncertainty in this thesis. In simulations with common random numbers, the same random numbers are used in the same sequence for the designs being compared. This is done to simulate the different designs under the same uncertain conditions, and the same is performed for shared uncertainty.

Considering shared uncertainty changes the criterion used to apply the elimination principle. Previously, the decision criterion was: if $\underline{U}(A) > \overline{U}(B)$ then eliminate B. With this criterion, each bound is computed unintentionally with separate instances of the same uncertainty variable, thus the bounds are overly conservative. In applying the knowledge of shared uncertainty variables, first the decision criterion needs to be reformulated to an equivalent criterion: if $\forall Z : \overline{U}(B,Z) - \underline{U}(A,Z) < 0$ then eliminate *B*, where $Z$ is a shared uncertain parameter. This criterion is best explained through a simple graphical example.

Assume that the performance of alternatives *A* and *B* varies with the one uncertain parameter as shown in the left of Figure 38. Based on this variation, the bounds on each alternative’s utility are shown in the top right of the figure. Since these bounds overlap significantly, elimination is not possible. However, elimination is possible in this example if shared uncertainty is considered. Considering shared uncertainty, the relative performance of the two alternatives is computed, and the result is shown in the two bottom plots. The entire interval of the difference between alternative $A$ and alternative $B$ is greater than 0. The criterion for elimination ( $\forall Z : \bar{U}(B,Z) - \underline{U}(A,Z) < 0$ ) is met. Alternative $A$ out-performs alternative $B$ in all uncertainty conditions, and alternative B can be eliminated; without considering shared uncertainty, this elimination is not possible.

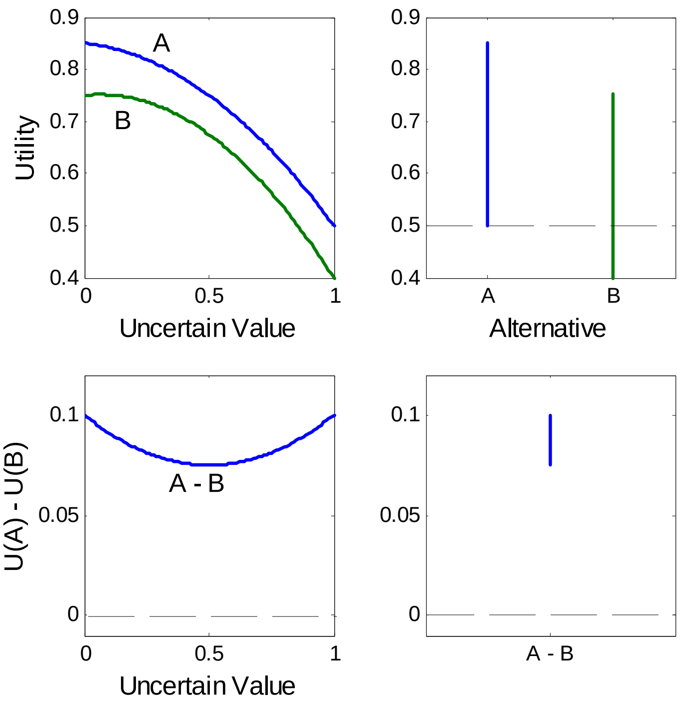



***Figure 38: Comparing two alternatives with and without considering shared uncertainty — only by considering shared uncertainty can alternative B be eliminated.***

In considering shared uncertainty, just as without it, there is a reference design against which the other designs are compared for elimination. While testing whether $\forall Z : \bar{U}(B,Z) - \underline{U}(A,Z) < 0$, the *B* design is being considered for elimination while the *A* design is the reference design. Selecting the appropriate reference alternative(s) is important in properly applying shared uncertainty toward elimination.

As seen in Figure 4.11, the performance of designs varies differently with respect to the uncertainty; due to this difference, different reference designs allow different designs to be eliminated. For example, consider four alternatives, all having performance affected by the single uncertain parameter, *Z*. The performance of each alternative with respect to that uncertain parameter is shown in the left half of Figure.39. Based on this figure, one would probably select alternative *A* as a reference design since it has the highest upper and lower bounds. Alternative *A* serves well as a reference design for eliminating alternative *B*, but when the Z parameter value is 1, both alternative *C* and *D* dominate alternative *A*. If one wants to proceed further in eliminating, alternative *C* has to be a reference as well. Alternative *C* dominates alternative *D*, thus allowing alternative *D* to be eliminated. Even though alternative *C* has lower performance bounds than alternative *A*, using alternative *C* as a reference design allows further elimination. This demonstration shows how using multiple reference designs aids eliminating.

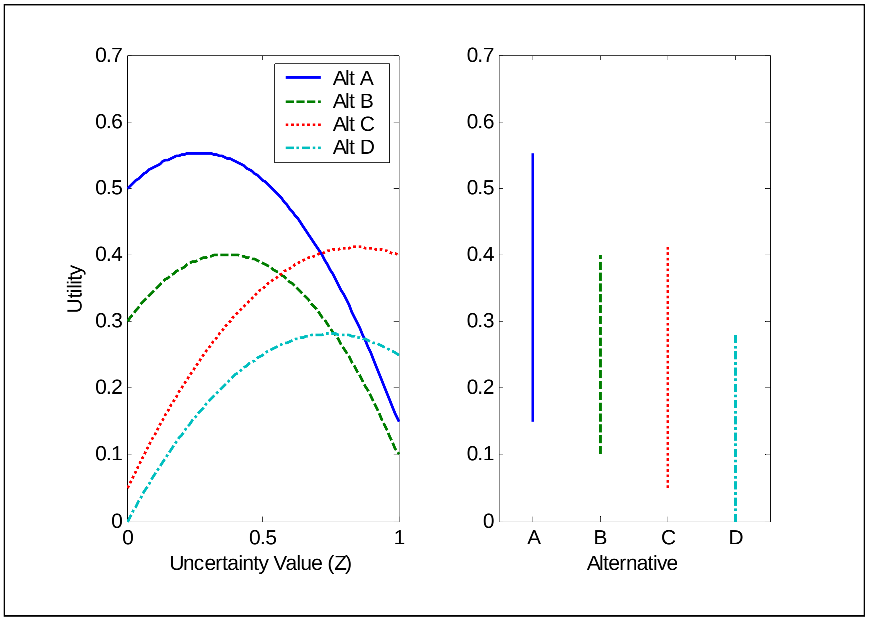



***Figure.39: Alternatives B and D can be eliminated by considering both A and C as references for comparison***

For eliminating, all designs that have not been eliminated should be used as reference designs. This ensures that the remaining designs are not dominated. Obviously, one should first use the reference design that eliminates the most designs from consideration; however, the specifics of how to do so are left for future work. Using multiple references allows the maximum elimination possible, taking full advantage of shared uncertainty.

Using a reference design in comparing alternatives for elimination also offers the advantage of reducing uncertainty about unspecified design variables. Often these unspecified design variables have a significant impact on the performance of a design. However, when design alternatives are compared for elimination these other design variables are not specified. Since these variables have a significant impact on design performance, this practice introduces significant uncertainty into the bounds on design performance that can make elimination more difficult. However, using a detailed, specific reference design eliminates much of this uncertainty and allows more elimination. Thus, the reference design chosen to use in elimination should be as specific as possible.

The same advantage offered by shared uncertainty does not apply to uncertainty that is not shared, therefore one must distinguish uncertainty as shared or not. Shared uncertainty must be the same for all the designs being compared; it must be from the same uncertainty source and the same instance that would be experienced by every design being compared. Otherwise, the uncertainty is not shared. This distinction in the uncertainty is necessary in applying shared uncertainty.

Considering shared uncertainty allows more designs to be eliminated. To employ this in the elimination, the difference in performance is used in the criterion to make the uncertainty shared by the designs explicit and easier to manage when considering all the uncertain conditions. The previous elimination criterion, which compares $\bar{U}\left(A_j\right)<\underline{U}\left(A_i\right)$, can be arranged to the equivalent: $\bar{U}\left(A_j\right)-\underline{U}\left(A_i\right)<0$. Then one must distinguish between shared uncertainty, $z_s$, and uncertainty that is specific to each alternative, $z_i$ for alternative $A_i$. Dominance between two alternatives, $A_i$ and $A_j$, is expressed as:

$$
\begin{gathered}
A_i \succ A_j \Leftrightarrow \\
\forall z_s \in Z_s, \forall z_i \in Z_i, \forall z_j \in Z_j : U\left(A_i, z_i, z_s\right) - U\left(A_j, z_j, z_s\right) > 0
\end{gathered}
$$

Or equivalently:

$$
\max_{\substack{z_s \in Z_s \\ z_j \in Z_j \\ z_i \in Z_i}} \left(U\left(A_j, z_j, z_s\right) - U\left(A_i, z_i, z_s\right)\right) < 0
$$

For alternative $A_j$ to be dominated, this must be true for at least one $A_i$. Or conversely, $A_j$ is non-dominated if there does not exist a single dominating $A_i$. To ensure that $A_j$ is non-dominated, the performance of the reference design alternative should be maximized: $\max_{A_i \in D} U\left(A_i, z_i, z_c\right)$. This leads to the final criterion for elimination under shared uncertainty that appears in Figure 40:

$$\max_{z_c \in Z_c, z_j \in Z_j, z_i \in Z_i} \left( U\left(A_j, z_j, z_c\right) - \max_{A_i \in D} U\left(A_i, z_i, z_c\right) \right)$$

The criterion, $\max_{z_c \in Z_c, z_j \in Z_j, z_i \in Z_i} \left( U\left(A_j, z_j, z_c\right) - \max_{A_i \in D} U\left(A_i, z_i, z_c\right) \right)$, can be simplified for easier computing. The best reference alternative can be moved to the front of the criterion to get: $\max_{z_c \in Z_c, z_j \in Z_j, z_i \in Z_i} \left( \min_{A_i \in D} \left( U\left(A_j, z_j, z_c\right) - U\left(A_i, z_i, z_c\right) \right) \right)$. This leaves the difference in the utility functions, $U\left(A_j, z_j, z_c\right) - U\left(A_i, z_i, z_c\right)$, to compute over the uncertain region. The difference in the utility functions is often monotonic with respect to the uncertain variables. In this case, computing the bounds is much easier; the performance bounds with respect to uncertainty occur at the boundary of the uncertainty. This was already demonstrated for bounding the performance of a single design; here it is demonstrated for bounding the performance between two designs in Figure 41. This figure shows two performance functions and their difference function. While the individual performance functions are not monotonic, the difference between the functions is monotonic.

*Consider design space D. Let* $A_i \subset D$ *and* $z_c \in Z_c$ *is shared uncertainty that is experienced by all of D and* $z_i \in Z_i$ *is uncertainty that is specific to* $A_i$.

***Elimination Criterion***

*Eliminate* $A_j$ *if and only if* $A_j$ *is dominated by at least one other design:*

$$\exists A_i \in D : A_i \succ A_j$$

*or:*

$$\max_{z_s \in Z_s, z_j \in Z_j, z_i \in Z_i} \left( U\left(A_j, z_j, z_s\right) - \max_{A_i \in D} U\left(A_i, z_i, z_s\right) \right) < 0$$

***Applying the Principle***

*All alternatives in the set* D *should be eliminated in accordance with the elimination criterion such that the remaining alternatives form a non-dominated solution set* $S \subset D$ *with:*

$$\forall A_i, A_j \in S : A_i \not\succ A_j$$

**Figure 40: The general elimination criterion for considering shared uncertainty**

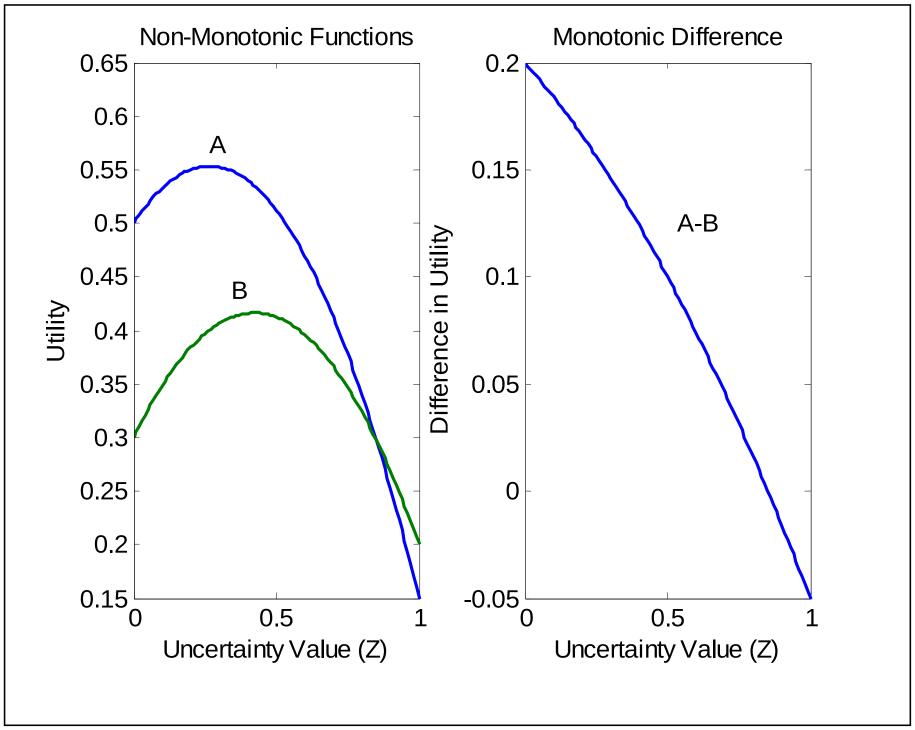



***Figure 41: The performance of the two alternatives are not monotonic with respect to the uncertain parameter, (left); however, the difference in performance between the two alternatives is monotonic (right)***

This example again demonstrates how monotonic functions are easy to bound while also demonstrating how one must be careful in using the monotonic assumption. In the example, the difference of two non-monotonic functions, $U(A,z)$ and $U(B,z)$, results in a function, $U(A,z)-U(B,z)$, that is monotonic. As demonstrated in the example, non-monotonic utility functions will not necessarily produce a non-monotonic difference between those utility functions and vice versa. The monotonic assumption must be evaluated for each function individually.

Therefore, to compute the bounds for eliminating, one should determine if the difference between the designs compared is monotonic with respect to uncertainty. If it is, then the bounds on performance can be computed by calculating utility at the different bounds of uncertainty. These utilities then can be examined to determine the bounds.

If the difference between the two alternatives is not monotonic then a more complicated method of searching the uncertainty is necessary. This is not covered in this section; rather, it is left as future work and studied more in depth in Chapter 6. Regardless of the method one employs in bounding, one should include shared uncertainty in computing those bounds.

Shared uncertainty can be extremely useful in eliminating and is incorporated in the elimination criterion later in this section. To demonstrate the effectiveness of considering shared uncertainty, attention is turned back to the beam example from the previous section.

### Beam Example

In the beam example, further eliminating was not possible because of the large bounds on design performance. To eliminate more designs, the shared uncertainty is taken into account by computing the relative performance. The Square Wood beam is used as the first reference. This reference was chosen because it has the highest lower-bound, and thus served as a good starting point in elimination. To compute the bounds on the relative performance, it was assumed that the difference function between any two alternatives is monotonic with respect to the uncertainty. This monotonic relation is shown in Figure 42 for the difference between the square wood beam and the square steel beam over the uncertain distance.

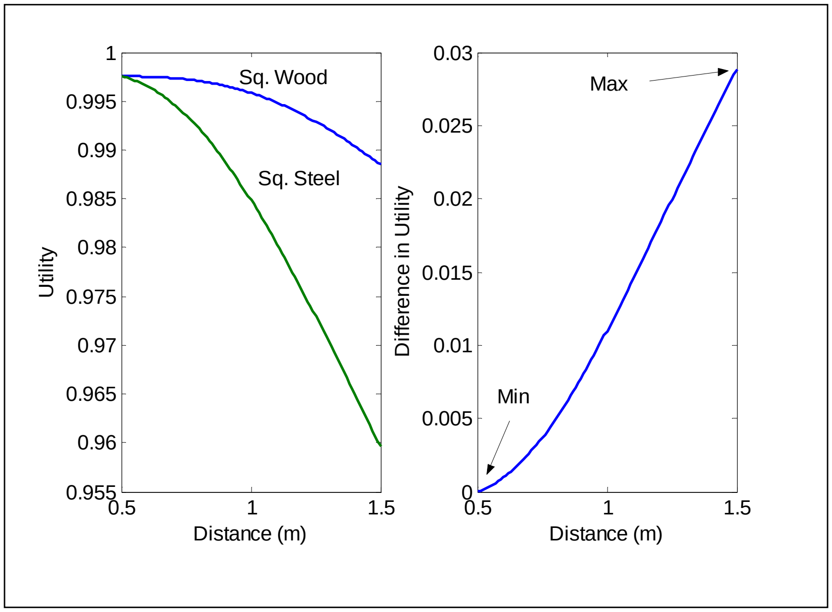



***Figure 42: The difference in performance between the square wood beam and the square steel beam is monotonic***

The difference functions between the all alternatives remaining are monotonic, thus the bound on the performance difference are computed by comparing the performance at the uncertainty bounds. The bounds on relative performances are shown in Figure 43 for all designs. From the results shown in Figure 43 one can eliminate the Aluminum Beam from consideration, as it never performs as well as the square wood beam. This elimination is shown in the overall process and design space in Figure 4.13. While both alternatives with aluminum have been eliminated in this branch of the design method, these alternatives would have been eliminated in the previous branch had a detailed reference design been used to compare them for elimination.

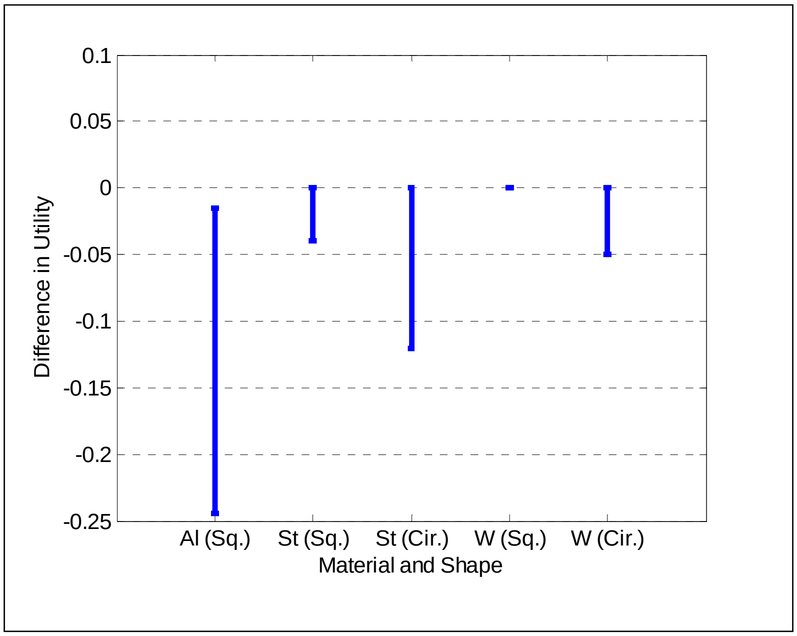



***Figure 43: The performance of the all the remaining design alternatives with respect to the square wood beam***

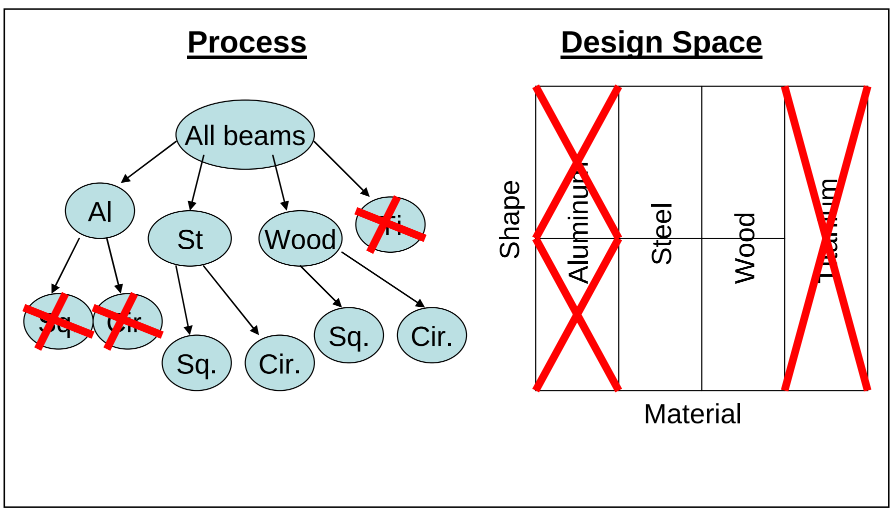



***Figure 44: The design process after all of the possible elimination with that reference design.***

Unfortunately, dominance cannot be established between the remaining beams based on this reference design; however, relative performance still can be used to further eliminate by using another reference design. As explained earlier, another reference design performs differently under the uncertain conditions and hopefully allows more designs to be eliminated. For this example the circular wood beam was chosen as a reference with the understanding that it would vary in a manner similar to the circular steel beam. Both of these beams have increased performance when the force is lower; in these conditions, less material in their cost-effective shape allows the beam to have higher utility. The resulting difference in utility from this is shown in Figure 45.

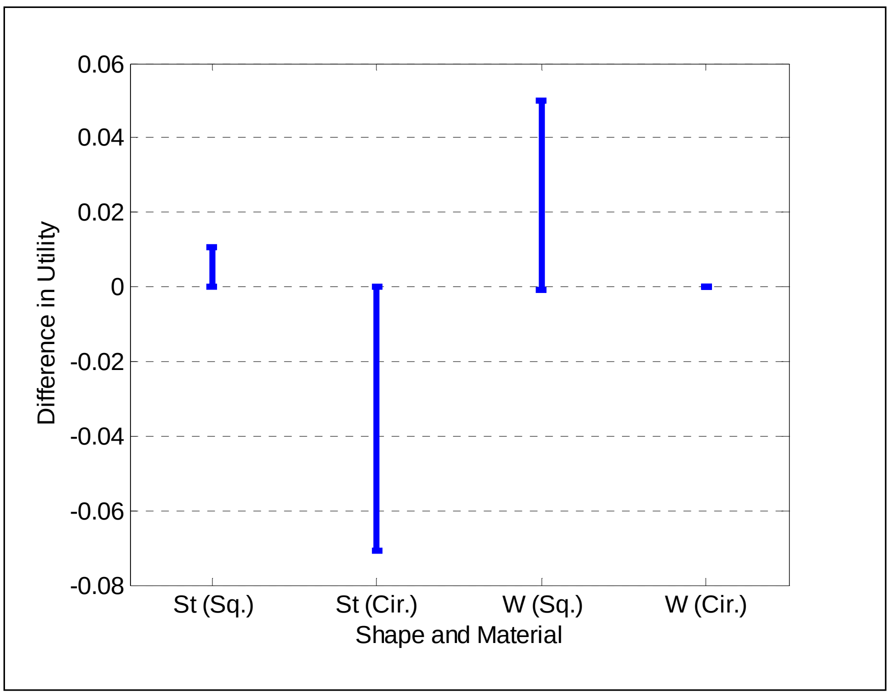



***Figure 45: The performance of the remaining design alternatives with respect to the circular wood beam***

From the performance bounds shown in Figure 45, one can eliminate the steel circular beam from consideration (the upper bound is slightly below 0). Since the performance of the circular shape beams vary in a similar manner, the superior circular beam is used to eliminate the other circular beam. This elimination is not possible if the square wood beam is used as a reference.

After elimination based on Figure 45, the design space appears as diagrammed in Figure 4.16. The remaining designs are non-dominant, and cannot be eliminated based on the given information. One must either gather additional information, or select an alternative without knowing whether it is the best alternative. Ideas about how this selection should be performed are presented in Chapter 6, as future work.

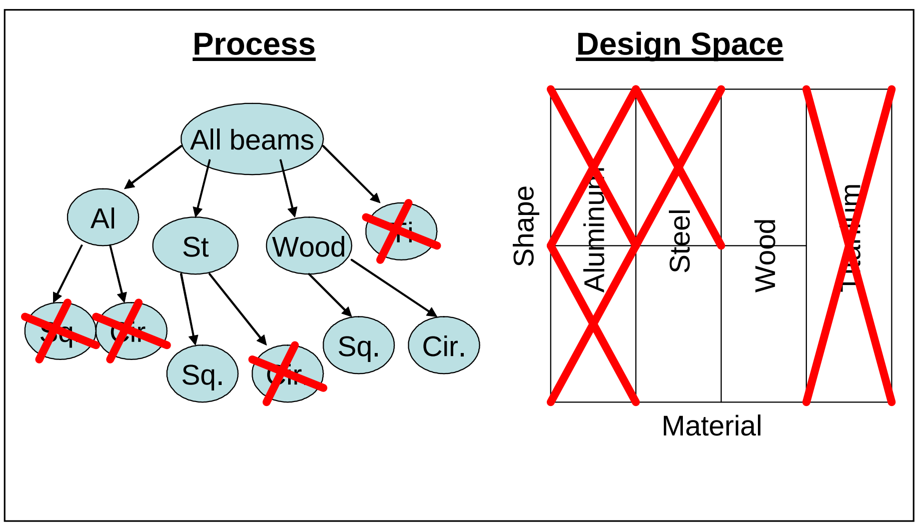



***Figure 46: The design process and space after eliminating with the circular wood beam. No more elimination is possible.***

This section of the thesis has demonstrated how knowledge about shared uncertainty can be applied to allow more elimination. Namely, shared uncertainty should be incorporated by eliminating based on relative performance; the relative performance should be computed with multiple designs as a reference. Applying this knowledge allows one to eliminate more designs.

While applying shared uncertainty allows more elimination, one may have to compare the detailed design alternatives in order to eliminate. This was the case with the aluminum beams. Evaluating all of these alternatives in details could be an option with design problems that involve only a few detailed alternatives; however, in more complicated systems, the number of alternatives to be evaluated grows exponentially with the number of design variables. Evaluating all of these alternatives could incur a prohibitive cost. To avoid this, one would like to eliminate designs at a higher level in the tree. This raises the questions: could these alternatives be eliminated with considering each one in detail? This is the subject of the next section.

### 4.3.2 Improving Elimination with a Specific, Detailed Reference Design

As pointed out in the previous section, one may have to compare detailed alternatives to eliminate more inferior design alternatives. While this approach is effective at eliminating alternatives, an approach that does not require all the alternatives to be detailed would be preferred. Such an approach is possible by using a specific, detailed reference design, which reduces the uncertainty about the unspecified design variables and allows more elimination without specifying those variables.

When one bounds the possible performance from an alternative, one must consider all the possible alternatives for the future design decisions. That is one bounds the performance of the family of design solutions. Because this family can include both good and bad design solutions, the bounds on that family must include these solutions and can be large. This concept is displayed in Figure 47. In this figure, the S2 subset of designs has large performance bounds because it includes S2.1, S2.2, S2.3, and S2.4.

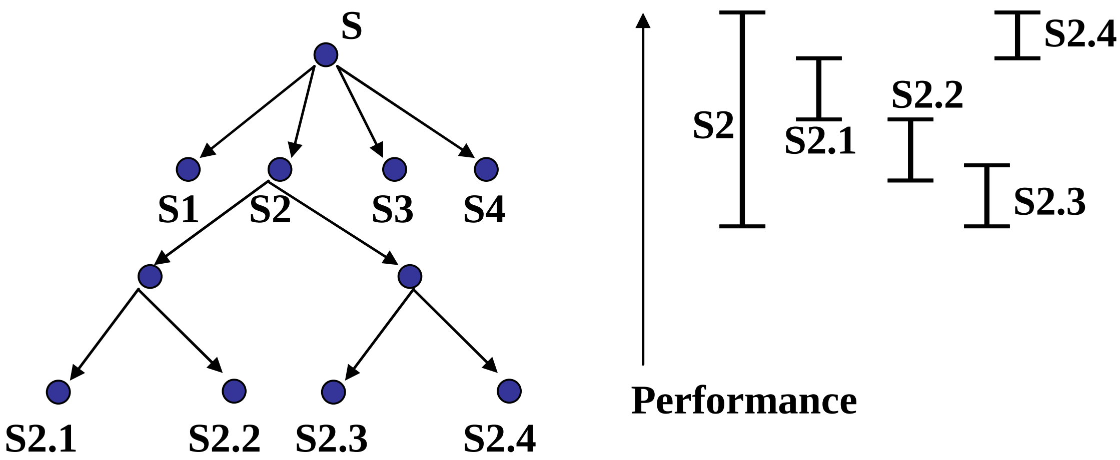



***Figure 47: The bounds on a design family include all members of that family. This results in large bounds on the family.***

These large bounds on a family of designs can limit elimination. If each family of designs considered for elimination includes designs that vary significantly in performance then the bounds on each of the design families will be large. Large bounds often limit one's ability to eliminate. These large bounds are shown for 4 families of designs in Figure 48. Based on these bounds, no elimination is possible.

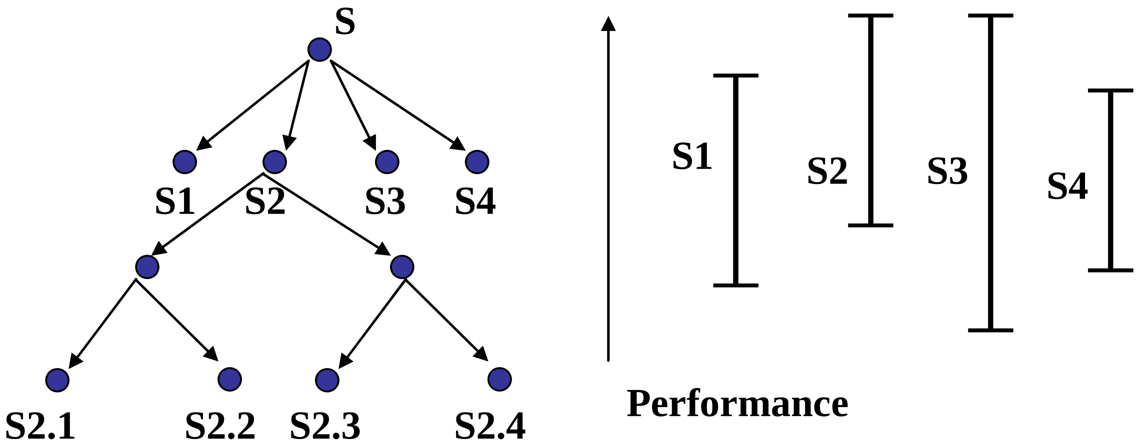



***Figure 48:* The bounds on four families of designs. Because these bounds include all members of each family, elimination is not possible.**

Elimination is possible however, if one uses a specific, detailed reference design. One may notice that the detailed design alternative, S2.4, has bounds that are on the upper end of the S2 bounds. In fact, the bounds on S2.4 dominate S1 and S4, as shown in Figure 49. So, by using S2.4 as a reference design, more elimination was possible. This is because S2.4 is detailed and does not include uncertainty about design decisions, which results in tight bounds on its performance.

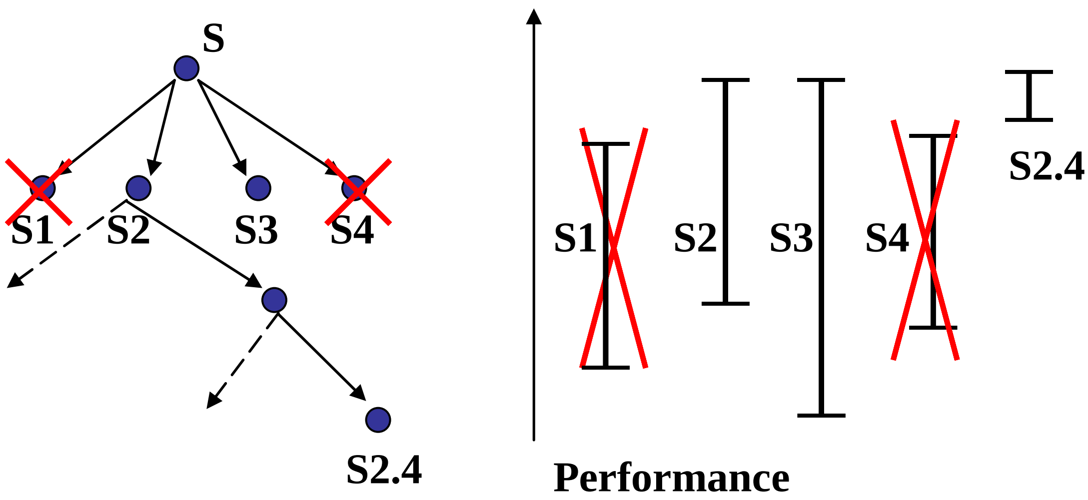



***Figure 49: If the detailed design alternative S2.4 is used as a reference design then the design families S1 and S4 can be eliminated. This elimination is not possible otherwise.***

This example demonstrates how one detailed, specific reference design can allow elimination of entire families of inferior designs without evaluating the particular instances of those families. That is, by detailing one design alternative to use as a reference design, there may not be a need to detail any more design alternatives. While this is a significant result, it does not change the formulation of the elimination criterion. Instead, the criterion is still as formulated in Figure 40; however, in this criterion, the reference design $A_i$ should be as detailed as possible.

There are some different options for obtaining this detailed reference design. First, one could use a previous design solution from a similar design problem. This provides a quick reference design that may be valuable in elimination. Second, one could perform a depth-first search to obtain a reference design that is then used in elimination. While this approach requires more resources it may be necessary when no previous designs exist. Lastly, one could arbitrarily select a design. This is the quickest method of obtaining a reference design in situations where there are no previous design solutions. Of course, these detailed reference design could be updated throughout the design process, as more knowledge is gained.

While this detail of elimination can be very useful in designs with multiple decisions, in the beam design there are just two decisions, so using a detailed, specific reference design has limited impact on elimination. For this reason, this detail of the elimination criterion is not demonstrated on the beam design example problem. For a demonstration of how the detailed reference design aids elimination, the reader should refer to Section 5.3.2, where this aspect of the elimination criterion is demonstrated on a gearbox design example.

## *4.4 Summary of Elimination under Interval-Based Uncertainty*

In this chapter, an elimination principle was developed, and knowledge about applying that elimination principle under uncertainty was introduced. The results of these previous sections are to be summarized and the integrated into the overall thesis picture in this section. In this section, the research question and hypothesis are revisited before examining how these fit into the thesis and validation strategy.

### 4.4.1 Revisiting the Research Question

The results from the previous section lead back to the primary research question and hypothesis. The primary research question and focus of this chapter is as follows:

**Q:** *Under conditions of interval-based uncertainty, how should one eliminate designs?*

Throughout the sections of this chapter, the hypothesis has been developed. for this question is:

**H:** *One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design and by accounting for shared uncertainty.*

This hypothesis is re-enforced in the previous sections of this chapter. In section 4.1, it was argued that at the very least, one's decisions should be consistent with their preferences; an idea consistent with utility theory (von Neumann and Morgenstern 1944; Luce and Raiffa 1957; Keeney 1974; Keeney and Raiffa 1993). To quantify these preferences one determines the utility of the designs attributes; the best design that should be selected will have the highest utility. However, under interval-based uncertainty the elimination is not that simple. This interval uncertainty leads to the following decision principle:

**Elimination Principle** One should eliminate a design alternative if, and only if, that design alternative is dominated throughout all uncertainty conditions by at least one other design alternative.

In using this decision principle, one may not be able to eliminate. To allow more elimination, knowledge about shared uncertainty should be applied, as explained in section 4.3. The basis for shared uncertainty lies in common random numbers, an approach already in application in simulation (Law and Kelton 2000), and interval analysis (Moore 1979; Alefeld and Herberger 1983; Kearfott and Kreinovich 1996; Hansen and Walster 2004). In considering shared uncertainty, the designs are compared for elimination under the same conditions. This allows more designs to be eliminated, sometimes significantly improving elimination.

Additionally, in applying this principle one should compare the design alternative to be eliminated with a reference design that is as detailed as possible. This limits the uncertainty introduced by unspecified design variables and allows elimination of inferior families of designs without going into the details of those families.

The contributions elaborated in this chapter were shown to be structurally sound. These contributions still must be fit in the validation strategy and thesis roadmap.

### 4.4.2 Chapter's Contribution to the Thesis

This chapter introduces and establishes the theoretical soundness of the work in this thesis. Specifically, the principle for elimination under interval-based uncertainty was presented and supported; this shows the hypothesis to be internally consistency, as pointed out in the validation strategy presented in Table 11.

As shown in this table, this chapter completes the theoretical structural validity that began in Chapters 2 and 3. In these chapters, the theoretical foundation for elimination under uncertainty was presented and the need for such a process was recognized. The foundation from Chapters 2 and 3 was used in Chapter 4 to show the internal consistency of the hypothesis.

**Table 11: Strategy for Validation**

| Quadrant of Validation Square | Thesis Chapter/Section | Aspect of Validation (for which Hypothesis) |
|---|---|---|
| | Chapter 2 | Literature Search to establish theoretical basis for proposed method |
| Theoretical Structural Validity (1) | Chapter 3 | Concept for Branch and Bound approach to design to set context for elimination |
| | **Chapter 4: Section 4.2** | **Theoretical structural soundness of rational decision-making** |
| | **Chapter 4: Section 4.3** | **Soundness of applying common uncertainty to elimination** |
| Empirical Structural Validity (2) | Chapter 5: Section 5.2 | Example's capability to test elimination method is established |
| Empirical Performance Validity (3) | Chapter 5: Section 5.5 | Example is evaluated to determine if design method is useful |
| | Chapter 5: Section 5.6 | Example decisions examined to verify performance of elimination method |

| Theoretical Performance Validity (4) | Chapter 6: Section 6.1 | Hypothesis is revisited and examined for validity to make 'Leap of Faith' |
| --- | --- | --- |

To continue in the validation strategy the performance of the work must be checked, this begins with the Empirical Structural Validity and Empirical Performance Validity in the next chapter with the use of an example problem.

Before jumping into the next chapter, the thesis roadmap is revisited in Figure 50 to give the reader perspective on what lies ahead. Attention is now turned toward Chapter 5, and the example problem presented within as the basis for empirical validity.

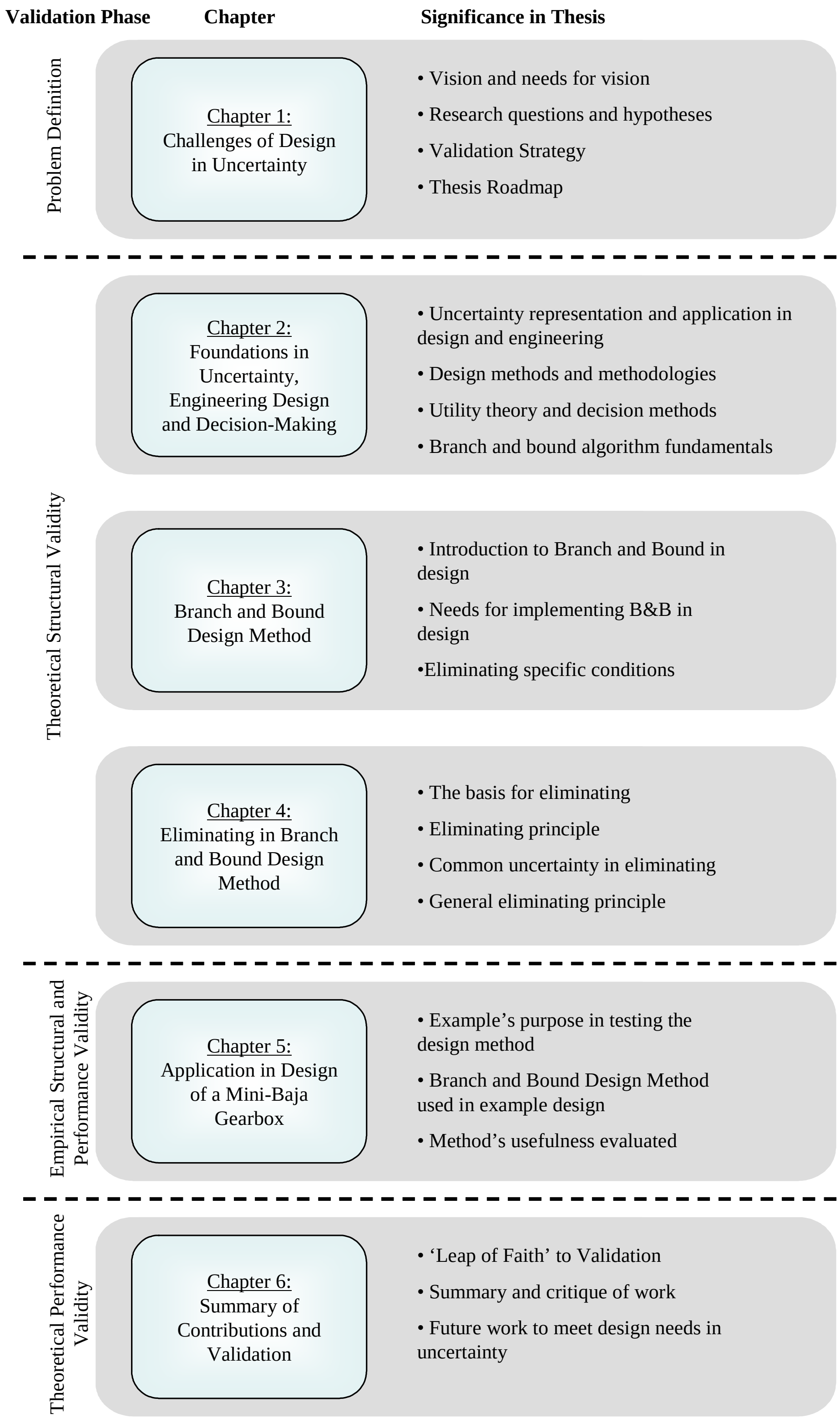



***Figure 50: Thesis Roadmap***

# CHAPTER 5

# ELIMINATION IN THE DESIGN OF A GEARBOX FOR A MINI-BAJA SAE CAR

In the previous chapter, the principle for elimination under interval-based uncertainty was derived, showing the hypothesis to be internally consistency and theoretically sound. However, the elimination method has not yet been shown to be useful; this is the next step in validation and the purpose of this chapter.

In this chapter, I test the hypothesis for its usefulness through the gearbox design for a mini-baja car. In Section 5.1, the example is introduced and the modeling of the uncertainty and system performance is explained. I then discuss specifically how this example tests the usefulness of the elimination method. The example is implemented with the elimination principle in Section 5.2, and the results of this example are discussed. In Section 5.3, I examine how the specific aspects of the elimination method allow for designs to be eliminated. Then in Section 5.4, I tie this chapter into my thesis, stating how the chapter contributes to validating my hypothesis.

## *5.1 Introduction to Mini-Baja and the Design Example*

In this section, the example design problem is first introduced and explained to give the reader context. Then I explain how the design is modeled. This includes models for the uncertainty, models for the performance, and models of my preferences for the car. I finish this section by pointing out the particular aspects of my work that are validated with the example design.

### 5.1.1 Design Example Description

The example design is a gearbox used in an SAE (Society of Automotive Engineers) Mini-Baja vehicle. This design is chosen because it has multiple attributes of concern, models for all of the attributes, and a practical application. This makes it a good choice to test the elimination method for practicality. Before testing the method, a context for the design is given by explaining how this sub-system fits into the overall vehicle.

SAE Mini-Baja competition consists of three regional and four international student competitions to build an off-road vehicle; an example of a Mini-Baja vehicle is shown in Figure 51. The students, with their vehicles, compete against each other in different events on rough, off-road terrain. Through these competitions the students gain a practical understanding of how their engineering can be applied, fostering innovation and competition among future automotive engineers.

To make the races more competitive, each of the teams must build their vehicle around the same Briggs & Stratton, Intek Model 20 engine (Engineers) 2004). The factory engine is governed by ignition-retarding at 4000rpm and produces 10 horsepower, as shown in the factory torque and horsepower curves in Appendix A. However, for Mini-Baja competition, the engine is limited to 3600rpm by restricting the fuel supplied to the engine (Engineers) 2004), thus the old torque curve is no longer valid. A new torque curve must be obtained experimentally by use of the team’s dynometer. The dynometer is rated for use with much more powerful engines, therefore the experimentally obtained data points given in Appendix A have significant

***Figure 51: An SAE Mini-Baja car in construction.***

uncertainty. This uncertainty is incorporated into the model used for predicting engine torque at a given speed, also contained in Appendix A. Because the fuel supply to the engine is restricted to limit the speed, the power of the engine is greatly reduced.

Since the power from the engine is so limited, proper tuning of the drivetrain is necessary to get the best performance possible from the engine. A diagram of the components in the drivetrain is shown in Figure 52. As shown in this figure, a Continuously-Variable Transmission (CVT) is used to maximize the performance. This transmission has a ratio of input rotational speed to output rotational speed that continuously adjusts to transmission ratios between 3.83 and

0.73. When the CVT is properly tuned, the CVT changes its ratio such that the engine remains at the speed that gives maximum power. The CVT also acts as a clutch in the drivetrain. Until the engine exceeds 1000 rpm, the belt on the CVT is not engaged and only minimal torque is transmitted.

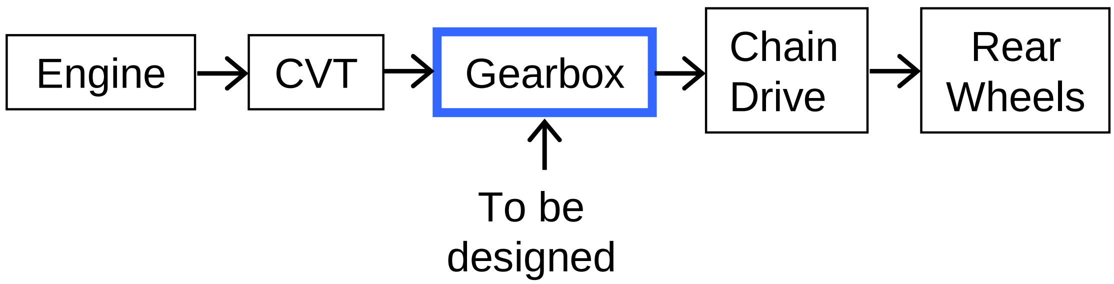



***Figure 52: Drivetrain components of SAE Mini-Baja Car***

If only the CVT is used to increase torque, a maximum of 16 N*m of torque reaches the wheels of the car. This is not enough torque to adequately accelerate the car on the off-road courses faced in competition, so a gearbox and chain-drive are used to increase the torque. Because the torque needs to be increased significantly, just one of these components in the drivetrain has to be very large. A drivetrain design with the chain-drive alone requires an output sprocket that is too large and heavy to fit on the rear driveshaft, while one with the gearbox alone requires gearing that does not fit in the allotted space under the engine, does not reach the rear driveshaft, and is too heavy. As a solution to this, the mini-baja team designed a chain-drive with a ratio of 2.73, but left the gearbox to be designed. This gearbox design is used as the example for illustrating the Branch and Bound design approach and the corresponding elimination method.

The setup for the gearbox has already been selected. The gearbox input and output shafts must rotate in the same direction, thus the three-gear setup shown in Figure 53 is used. In this setup the specifics of the gears have not been decided; this is the purpose of the example design. The example design involves the selection of five design variables that specify the gears in the gearbox; these variables are as follows:

***Gear Width [cm]:*** This continuous variable determines how wide the teeth are on every gear. Different gear widths for the three different gears always results in an inferior design, so it is decided that the width is the same for every gear.

***Gear Module [mm/tooth]:*** This continuous variable determines how many teeth per millimeter are on every gear. Different gear modules for different gears results in a design that does not function, so it is decided that the width is the same for every gear.

***Input Gear Diameter [cm]:*** This continuous variable, in combination with the module, specifies the number of teeth on the input gear. The number of teeth is not used because it is a discrete variable, and the design then is a mixture of discrete and continuous variables, which is more difficult to keep track of in elimination. Additionally, when one bounds the performance of a subset of designs, one needs to consider the unspecified design variables in these bounds. When these unspecified design variables are a mixture of discrete and continuous design variables then the bounds are more difficult to compute. For simplicity, I use the gear diameters instead of the number of teeth on the gears. While the gear diameter was chosen because of practical considerations, one should use the number of teeth on the gear instead in the design of a gearbox.

***Idler Gear Diameter [cm]:*** This continuous variable, in combination with the module, specifies the number of teeth on the idler gear. The number of teeth is not used for the same reasons as stated for the input gear.

***Gear Ratio:*** This continuous variable, in combination with the input gear diameter, specifies the diameter of the output gear. The gear ratio is used in this situation because it correlates directly with the vehicle acceleration and maximum velocity.

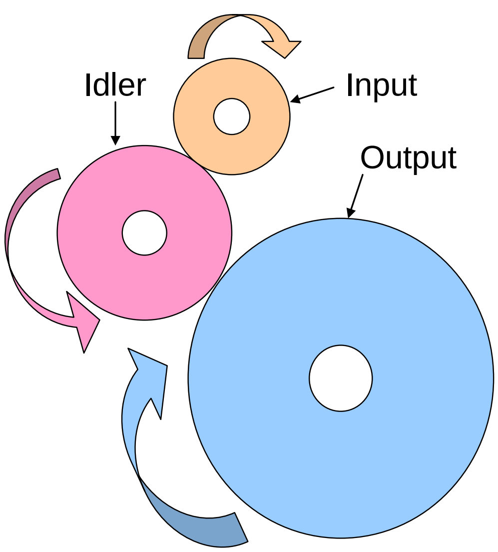



***Figure 53: Gearbox setup used in gearbox design.***

These design variables specify the gears used in the gearbox allow for the various gearbox performance attributes to be determined. The performance of each gearbox is judged on the basis of the following attributes:

***Vehicle Top Speed [m/s]:*** The maximum velocity that the vehicle can reach directly affects the car's race performance: higher top speed results in faster lap times.

***Vehicle Acceleration [m/s$^2$]:*** There are multiple measures that can be used for this attribute. Often cars are judged based on time to cover a specific distance, such as a quarter-mile, or their time to a specific speed, such as 60mph or 100km/hr. Although these provide an excellent measure of the car performance, they are unnecessarily complicated to model. Instead, the instantaneous acceleration at startup and maximum CVT ratio is used. A designer tries to maximize this measure of acceleration in designing the gearbox.

***Reliability [%]:*** The reliability of the gearbox to make it through two seasons of Baja Competition is evaluated. In two seasons, the car travels approximately 400 kilometers in rigorous competition and test-drives.

***Mass [kg]:*** Only the mass of the gears is considered in the mass of the gearbox, thus the mass of the gearbox is the sum of the three gear masses.

One may notice that the cost of the gears is not included. This is because the GT Mini-Baja team has a sponsorship with Rush Gears, who provide them with free gears. Thus, cost is not an attribute of concern in the design process, as it normally would be. To determine the performance of the remaining four attributes, the gearbox is modeled as explained in the next sub-section.

### 5.1.2 Design Performance Modeling

The design performance models translate the design variables chosen into design attributes, which are combined into a utility function. In this section, the models for these attributes are explained. For each of these attributes, it is assumed that the other aspects of the design – the shafts, bearings, and housing – do not affect performance sufficiently to be included.

### Vehicle Maximum Velocity

The maximum velocity of the vehicle is reached when the frictional forces equal the force produced by the drivetrain. Therefore, at the maximum velocity, the following equation holds:

$$F_{\text{friction}} = F_{\text{drivetrain}} \tag{5.1}$$

The force produced by the drive train comes from the engine torque, which is transmitted through the drivetrain to the wheels, where it is applied to the ground as force:

$$T_{\text{wheels}} = \rho_{\text{CVT}}\rho_{\text{chain}}\rho_{\text{gearbox}}T_{\text{engine}}\left(\omega_{\text{engine}}\right) \tag{5.2}$$

where $T_{\text{engine}}\left(\omega\right)$ is the engine torque at the engine speed, $\omega_{\text{engine}}$; $\rho_{\text{gearbox}}$ is the gear ratio of the gearbox being designed; $\rho_{\text{chain}}$ is the ratio of the chain drive; $\rho_{\text{CVT}}$ is the momentary ratio of the continuously-variable transmission (CVT). Since both the CVT ratio and engine speed are not known, these two unknowns make it difficult to compute the wheel torque with only the knowledge of the vehicle speed, as is needed in top speed calculations.

To simplify this calculation, it is assumed that the CVT ratio is tuned properly and the gear ratio of the gearbox is properly chosen. In this situation, I assume that the top speed occurs with the CVT at its minimum ratio of 0.73. This is equivalent to assuming an automobile reaches its top speed in its highest gear. With this assumption, Equation (5.2) is used to compute the wheel torque, which  is used to compute the force at the wheels using the following equation:

$$F_{\text{wheels}} = \frac{T_{\text{wheels}}}{r_{\text{wheels}}} \tag{5.3}$$

where $r_{\text{wheels}}$ is the radius of the wheels, $F_{\text{wheels}}$ is the force produced at the wheels. Equation (5.3) provides the force at the wheels for a given engine speed, however to compute the top speed, it is necessary to compute $F_{\text{wheels}}$ for a given vehicle speed:

$$\omega_{\text{engine}} = \frac{\rho_{\text{CVT}}\rho_{\text{chain}}\rho_{\text{gearbox}}V_{\text{vehicle}}}{2\pi r_{\text{wheels}}} \tag{5.4}$$

where $V_{\text{vehicle}}$ is the velocity of the vehicle, $\omega_{\text{engine}}$ is the engine speed, which can be used to compute the engine torque through the following equation:

$$T_{\text{engine}}\left(\omega_{\text{engine}}\right) = B_1\omega_{\text{engine}} + B_2\omega_{\text{engine}}^2 + B_3\omega_{\text{engine}}^3 \pm \varepsilon_{\text{torque}} \tag{5.5}$$

where $\omega_{\text{engine}}$ is the engine speed, $T_{\text{engine}}\left(\omega\right)$ is the engine torque, $B_1$, $B_2$, $B_3$ are engine parameters that were determined by linear regression on engine data as presented in Appendix A. Based on the results of this regression, it was determined that the model could predict engine torque within $\pm\varepsilon_{\text{torque}}$.

For a given vehicle velocity, Equation (5.4) is used to compute an engine speed; that engine speed is used to compute the force at the wheels using Equations (5.2), (5.3), and (5.5). Following this procedure, results in a force for a given vehicle velocity.

At the maximum velocity, the forced produced by the drivetrain is balanced against the friction force, therefore the friction force must be determined with respect to car velocity. The friction forces are produced both by friction in the drivetrain and air resistance on the outside of the vehicle. Separate models for each of these friction forces were determined empirically based on data. The resulting model for the internal friction was determined to be a linear relation between velocity and friction force, as follows:

$$F_{\text{internal}} = c_{\text{drivetrain}} V_{\text{vehicle}} \tag{5.5}$$

where $c_{\text{drivetrain}}$ is the coefficient of friction with the drivetrain, $V_{\text{vehicle}}$ is the vehicle velocity, and $F_{\text{internal}}$ is the internal friction force. The external friction is modeled to have a quadratic relation with velocity, consistent with drag coefficients in Fluid Dynamics (Munson 2001), as follows:

$$F_{\text{air}} = c_{\text{drag}} V_{\text{vehicle}}^2 \tag{5.6}$$

where $c_{\text{drag}}$ is the drag coefficient due to air movement on the vehicle external and $F_{\text{air}}$ is the friction due to air resistance. The friction from air resistance and the drivetrain are summed to get the total friction resisting further acceleration:

$$F_{\text{friction}} = F_{\text{internal}} + F_{\text{air}} \tag{5.7}$$

This friction force balances the force produced at the wheels by the force from drivetrain at the vehicle top speed, as given in Equation (5.1). The top speed is found by determining the velocity where these forces are balanced.

## Vehicle Acceleration

The instantaneous acceleration is computed at an engine speed of 1000rpm, which is just above the idling engine speed, and the maximum CVT ratio of 3.83. To compute this acceleration, the following equation is applied:

$$a_{\text{vehicle}} = \frac{\sum F}{m_{\text{vehicle}}} \tag{5.8}$$

where $a_{\text{vehicle}}$ is the instantaneous acceleration of the vehicle, $m_{\text{vehicle}}$ is the mass of the vehicle with driver, and $\sum F$ is the sum of the forces acting on the vehicle. The forces acting on the vehicle are the force of friction, $F_{\text{friction}}$, and the force from the drivetrain, $F_{\text{drivetrain}}$, at the wheels. These forces are summed in calculating the acceleration of the vehicle, as follows:

$$\sum F = F_{\text{friction}} + F_{\text{drivetrain}} \tag{5.9}$$

where $F_{\text{friction}}$ and $F_{\text{drivetrain}}$ are calculated using Equations (5.2) to (5.7).

## Gear Reliability

The reliability of the gearbox is computed as follows:

$$R_{\text{gearbox}} = R_{\text{input}} R_{\text{idler}} R_{\text{output}} \tag{5.10}$$

where $R_{\text{gearbox}}$ is the reliability of the gearbox, $R_{\text{input}}$, $R_{\text{idler}}$, and $R_{\text{output}}$ are the reliability of the respective gears. The reliability of these gears incorporates both the reliability against surface failure and the reliability against failure in bending; these are accounted for in Equation (5.11).

$$R_{\text{gear}} = R_{\text{contact}} R_{\text{bend}} \tag{5.11}$$

where, $R_{\text{gear}}$ is the reliability for a specific gear, $R_{\text{contact}}$ is the reliability against surface failure, and $R_{\text{bend}}$ is the reliability against failure in bending. These reliabilities are functions of the stress on each gear and strength of the gear both contact and bending. Each of these must be computed before the reliability can be calculated.

The stress in bending is computed using the equation given by the American Gear Manufacturer's Association (AGMA) (AGMA 1988; AGMA 1989):

$$\sigma_b = \frac{F_t}{WmJ} \frac{K_a K_m}{K_v} K_s K_B K_I \tag{5.12}$$

where $\sigma_b$ is the bending stress; $F_t$ is the tangential force on the gear teeth; $W$ is the width of the gear; $m$ is the module; $J$ is the dimensionless geometry factor that can be determined from the design variables; $K_a$ is the application factor that accounts for variations in the loading conditions; $K_m$ is the load distribution factor that is dependent on the width of the gears; $K_v$ is the dynamic factor that accounts for vibrations in the gears due to tooth velocity, which is dependent on the gear diameters; $K_s$ is the size factor that is dependent on gear size, $K_B$ is the rim thickness factor that incorporates stressed caused by a large gear hub; $K_I$ is idler factor, which accounts for the extra loading cycles on the idler gear. The process for finding these factors is explained for each by Norton(Norton 2000).

The stress that is calculated in Equation (5.12) is compared against the adjusted strength of the material. The strength of the gear material is adjusted by using Equation (5.13), taken from AGMA(AGMA 1988).

$$S_{fb} = \frac{K_L}{K_T} S'_{fb} \tag{5.13}$$

where $S_{fb}$ is the adjusted fatigue strength in bending; $K_L$ is the life factor of the gear, which is determined from the number of cycles in the life of the gear; $K_T$ is the temperature factor, and $S'_{fb}$ is the published bending fatigue strength. Once again, more detailed explanation of these factors is in Norton(Norton 2000).

With the bending strength and stress from Equations (5.12) and (5.13), the reliability factor is calculated as follows:

$$K_R = \frac{S_{fb}}{\sigma_b} \tag{5.14}$$

where $K_R$ is the reliability factor. This reliability factor is related to the reliability of the gear as is determined by the American Gear Manufacturer's Association(AGMA 1988) and given in (Norton 2000); in both of these references, the relationship was only given in terms of discrete points. To make this relationship useful, an expression for this relationship is developed using a best-fit of a Weibull distribution, which is commonly used in materials reliability to relate the nominal stresses to the reliability (pages 8-15 and 25-15 in (Nikolaidis, Ghiocel et al. 2005)); the details of this best-fit are available from the author. This relationship is shown in Figure 54 along with the discrete points given by AGMA and Norton. It is assumed that this fit is usable to relate the reliability factor to reliability.

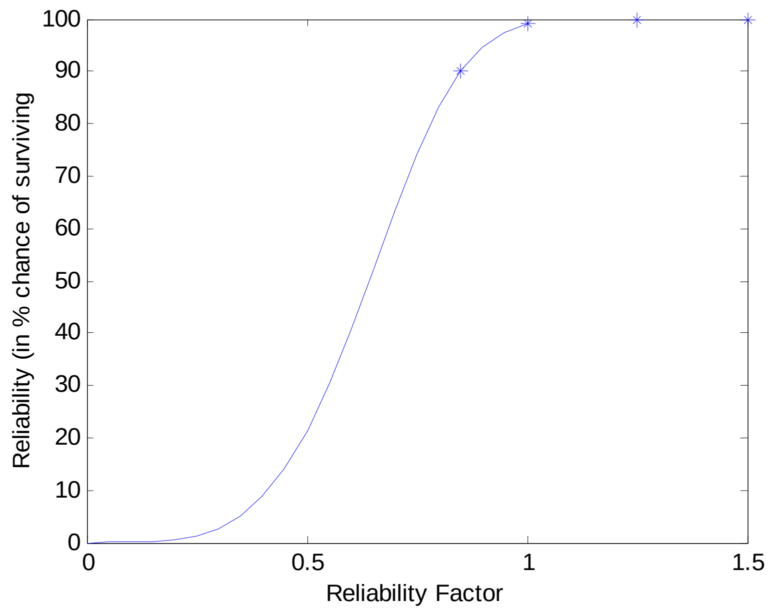



***Figure 54:   Reliability of the Gear as a function of the reliability factor***

The function shown in Figure 54 is used to convert the reliability factor to the actual reliability against failure in bending and is as follows:

$$R_{\text{bend}} = 1 - \exp\left(\left(\frac{-K_R}{0.699}\right)^{4.26}\right), \tag{5.15}$$

where $K_R$ is the reliability factor and $R_{\text{bend}}$ is the chance of the gear completing its lifetime without failing due to bending stress.

Through Equations (5.12-15), one computes the reliability against failure from bending stress, however in order to calculate the reliability of the gears, one must also compute the reliability against surface failure. To compute the reliability against surface failure, the surface stress of the gears is computed as follows (AGMA 1989):

$$\sigma_c = C_p \sqrt{\frac{F_t}{WId} \frac{C_a C_m}{C_v} C_s C_f} \tag{5.16}$$

where $\sigma_c$ is the contact stress; $C_p$ is the elastic coefficient; $F_t$ is the tangential force on the gear teeth; $W$ is the width of the gears; $I$ is the surface geometry factor, which is determined from the geometry of the gears; $d$ is the diameter of the gear; $C_a$ is the application factor; $C_m$ is the load distribution factor; $C_v$ is the dynamic factor; $C_s$ is the size factor; $C_f$ is the surface finish factor, which dependent on the quality of the gear manufacturing process. Again, these factors and the process for finding each factors are explained by Norton(Norton 2000). The reader should note that the $C_a$, $C_m$, $C_v$, and $C_s$ factors are equivalent, respectively, to the $K_a$, $K_m$, $K_v$, and $K_s$ factors.

The stress obtained from Equation (5.16) is compared against the adjusted contact strength of the material. The contact strength of the gear material is as follows (AGMA 1988):

$$S_{fc} = \frac{C_L C_H}{C_T} S_{fc}^{'} \tag{5.17}$$

where $S_{fc}$ is the adjusted fatigue strength in contact; $C_L$ is the life factor of the gear; $C_T$ is the temperature factor; $C_H$ is the hardness factor, which is a materials property; $S'_{fc}$ is the published contact fatigue strength. $C_T$ is equivalent to $K_T$ from Equation (5.13); $C_L$ serves the same purpose as $K_L$, but has a different value specific to contact stresses. For more detailed information on each of these factors, the reader should refer to Norton's explanation of each(Norton 2000).

The contact stress and strength, obtained respectively from Equation (5.16) and (5.17), are applied in Equation (5.18) to obtain the reliability factor.

$$K_R = \frac{S_{fc}}{\sigma_c} \tag{5.18}$$

where $K_R$ is the reliability factor. This reliability factor has the same meaning as the reliability factor for bending stress and is computed via Equation (5.15) as well.

### Gearbox Mass

Based on the given design variables, the exact mass of the gearbox is only known for a few of the gear configurations. Since this mass cannot be predicted for all possible gears, instead the mass of the gearbox is bound for the gears. These bounds are determined first for the individual gears and then summed to get the total mass of the gearbox. These costs are based on gear prices provided by Martin Sprocket & Gear, Inc. (Martin Sprocket & Gear 2003); the prices from Martin's catalog were modeled with respect to the different design variables using regression. The details of this regression are available from the author. This resulted in the following model:

$$M_g = c_1 w d^2 \tag{5.22}$$

where $c_1 = 0.0068\,\text{kg/cm}^3$, a constant in the model, $w$ is the width of the gear in centimeters, $d$ is the diameter of the gear in centimeters, and $M_g$ is the predicted mass of the given gear. This predicted cost of the gear varies from the actual cost by as much as 2.27kg below the actual cost and as much as 1.73kg above the actual cost. This uncertainty could be incorporated into the model, as given in Equation (5.23).

$$M_g = c_1 w d^2 + \varepsilon \tag{5.23}$$

where $\varepsilon = \left[-1.73,\, 2.27\right]\text{kg}$. However, these bounds are too broad for the gears considered and would limit the ability to eliminate design alternatives. To address this problem, different functions are used to predict the upper and lower bounds of the model. The upper bound on the mass is as follows:

$$M_{g,\text{upper}} = c_{\text{upper}} w d^2 + \varepsilon_{\text{upper}} \tag{5.24}$$

where $c_{\text{upper}} = 0.0077\,\text{kg/cm}^3$, $\varepsilon_{\text{upper}} = 0.023\,\text{kg}$ and $M_{g,\text{upper}}$ is the upper bound on the mass of the gear. Likewise the lower bound is computed:

$$M_{g,\text{lower}} = c_{\text{lower}} w d^2 + \varepsilon_{\text{lower}} \tag{5.25}$$

where $c_{\text{lower}} = 0.0064\,\text{kg/cm}^3$, $\varepsilon_{\text{lower}} = -0.037\,\text{kg/cm}^3$ and $M_{g,\text{lower}}$ is the lower bound on the mass of the gear. This separate functions for the upper and lower bounds actually result in larger bounds for large gear sizes; however, these gears are not likely to be used in the design because of their excessive mass. The smaller gears that are considered closely have much tighter upper and lower bounds with the functions given in Equation (5.24) and (5.25).

These upper and lower bounds are for the mass of a single gear; to determine the mass of the gearbox, the individual gear masses must be summed, as follows:

$$M_{\text{gearbox}} = M_{\text{input}} + M_{\text{idler}} + M_{\text{output}} \tag{5.26}$$

where $M_{\text{input}}$ is the input gear mass, $M_{\text{idler}}$ is the idler gear mass, $M_{\text{output}}$ is the output gear mass, and $M_{\text{gearbox}}$ is the total mass of the gearbox. The masses of the individual gears are computed as given in Equation (5.24) and (5.25), with interval bounds, and Equation (5.26) results in an interval bound on the gearbox mass.

## Overall Performance: Formulating Utilities

The four performance attributes need to be incorporated into some overall measure of performance that can be used under uncertain conditions. To meet this need, the independent performance attributes are formulated into a utility function.

To create this utility function, utility functions are formulated for each of the independent attributes and then combined in a single, overall utility function; this process is based on the procedure given by Keeney and Raffa for mutually independent utility functions, detailed in Section 2.4.1 (p. 219-281 in (Keeney and Raiffa 1993)). The independent utility function is then fit to these values and adjustments are made to the function as seen necessary by the decision-maker. This process is used to assess my preferences for the independent utility functions. The lottery points used and the best-fit functions obtained are available from the author. This process leads to the following attribute utility functions. For the acceleration, the utility is as follows:

$$U\left(a_{\text{vehicle}}\right) = 1 - \exp\left(-\left(\frac{a_{\text{vehicle}}}{2}\right)^{0.75}\right) \tag{5.27}$$

where $a_{\text{vehicle}}$ is the vehicle acceleration, $U\left(a_{\text{vehicle}}\right)$ is the utility from the acceleration. For top speed of the vehicle, the utility function is as follows:

$$U\left(v_{\max}\right) = 1 - \exp\left(-\left(\frac{v_{\max}}{14}\right)^{1.375}\right) \tag{5.28}$$

where $v_{\max}$ is the top speed, $U\left(v_{\max}\right)$ is the utility from the top speed. For the mass of the gearbox, the utility function is as follows:

$$U\left(M_{\text{gearbox}}\right) = \exp\left(-\left(\frac{M_{\text{gearbox}}}{5}\right)^{0.5}\right) \tag{5.29}$$

where $M_{\text{gearbox}}$ is the mass of the gearbox, $U\left(M_{\text{gearbox}}\right)$ is the utility from the mass.

The utility function for the reliability of the gearbox is different. The gearbox reliability is the probability of whether the gearbox will operate through its lifetime or not. When this attribute is compared to obtain a certainty equivalent, it correlates directly to the lottery values used in the test. That is one should be indifferent between an alpha chance at 100% reliability and having a reliability of alpha*100%. This is shown in Figure 55, where the equivalent certain value of the reliability is equal to the chance of having 100% chance of reliability.

Therefore, when the indifference lottery is applied to this reliability, the utility from reliability is equal directly to the decimal version of the reliability, as follows:

$$U\left(R_{\text{gearbox}}\right) = R_{\text{gearbox}} \tag{5.30}$$

where $R_{\text{gearbox}}$ is the reliability of the gearbox, and $U\left(R_{\text{gearbox}}\right)$ is the utility from the reliability of the gearbox. These attribute utility functions are shown versus their attributes in Figure 56.

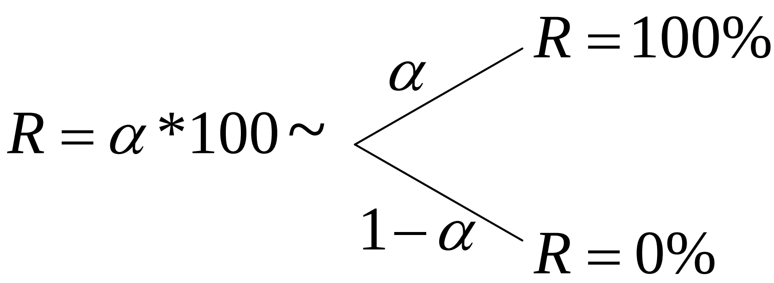



***Figure 55: One should be indifferent between a alpha chance at 100% reliability and having a reliability of alpha***

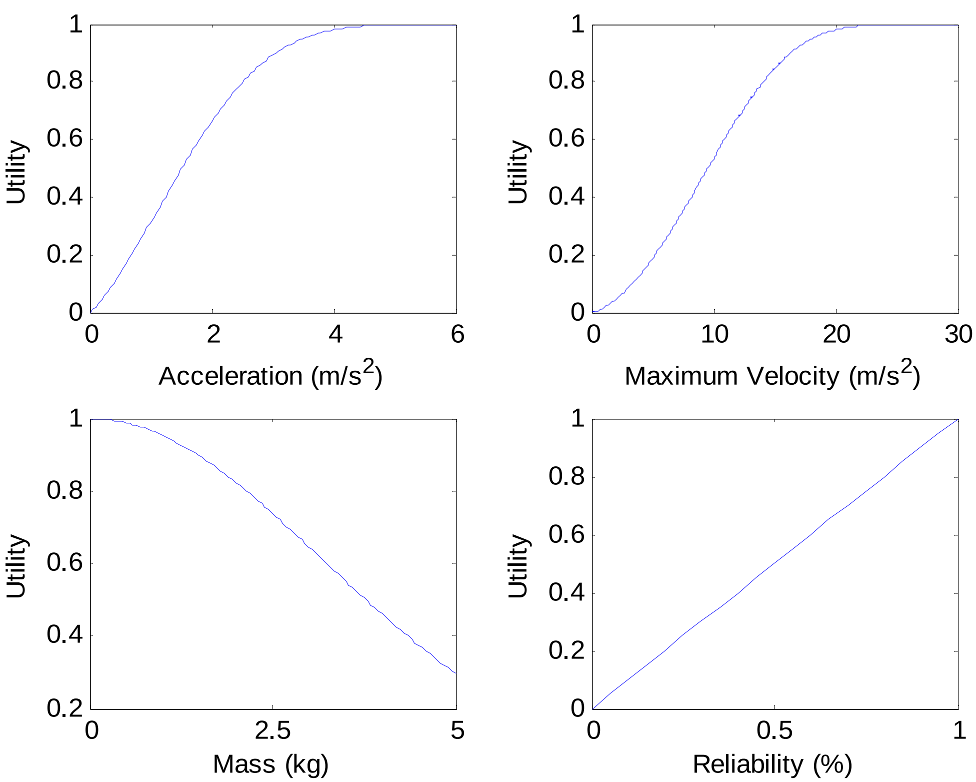



***Figure 56: Individual utility attribute functions for each attribute.***

The reader should take note of the utility function from the mass. The first function presented to bound the mass provided an interval of bounds approximately 8kg wide; however, such wide bounds results in bounds on the utility that could range by as much as 0.923. Based on this range of utilities, one has difficulty eliminating design alternatives. This is the important reason for changing the model used to bound the mass into separate equations for the upper and lower bounds.

The individual attribute utility functions are combined into a multiplicative utility function:

$$U_{\text{gearbox}} = U_{\text{accel}} U_{\text{speed}} U_{\text{mass}} U_{\text{rely}} \tag{5.31}$$

where $U_{\text{accel}}$ is the utility from the acceleration, $U_{\text{speed}}$ is the utility from the top speed, $U_{\text{mass}}$ is the utility from the mass, $U_{\text{rely}}$ is the utility from the reliability, and $U_{\text{gearbox}}$ is the overall utility of the gearbox. This multiplicative form of the utility function is equivalent to the typically applied additive form (p. 238 in (Keeney and Raiffa 1993)).

## Modeling Uncertainty

As pointed out in the modeling section, each model's uncertainty is incorporated into its results; however, this is not the only uncertainty considered. Uncertainty in the model parameters and the inputs to the model are incorporated into the results of the model. These uncertainty sources are given in Table 12 with bounds on their value. Some of these values were determined based on the limits of the models employed to calculate vehicle performance, while some of these bounds are based on data too sparse for a precise probability distribution. In these cases, the epistemic uncertainty about the distribution is larger than the aleatory uncertainty, thus an intervals is used to represent this uncertainty. For example, the bending strength of the gear material used is bounded based on results from three different materials properties tables (Ashby 1999; Norton 2000; Martin Sprocket & Gear 2003). Even though the bending strength of a material is normally distributed, the data is too sparse to determine the probability distribution for the materials strength, so an interval that represents this lack of knowledge about the distribution is used instead. While experienced designers would not likely have to rely on such sparse data, designers addressing a new domain may not have extensive information available to them.

Since the example design tests the effectiveness of considering shared uncertainty, the shared and independent sources of uncertainty are pointed out in the Table 12 as well. These uncertainties are included in the problem formulation given in Table 13.

**Table 12: Uncertainty in Mini-Baja Gearbox Design Problem**

| Source of Uncertainty | Uncertainty Representation | Explanation | Common Uncertainty |
|---|---|---|---|
| Driver Mass | [57, 102] kg ([125, 225] lbs) | Varies due to the different vehicle drivers | Shared Source |
| Car Mass | [180, 205] ([400, 450] lbs) | Varies based on the weight of the other component – it is assumed that the gearbox weight is not significant | Shared Source |
| Engine Torque | Equation – See Appendix A for details | Torque is not perfectly known, rather the torque-speed curve is bounded | Shared Source |
| Drag Coefficient * Area | [0.197, 0.348] $n/(m/s)^2$ | Varies based on the ambient air conditions | Shared Source |
| Coefficient of Internal Friction | [0, 0.01471] $n/(m/s)$ | Varies based on the performance of drivetrain components | Shared Source |

| | | | |
|---|---|---|---|
| Percentage of Mass on Rear Wheels | [0.50, 0.75] | Construction of the vehicle will change this characteristic | Shared Source |
| Coefficient of Friction with Road | [0.50, 1.00] | Varies based on terrain | Shared Source |
| Gear Quality | [7, 12] | Gear manufacturing affects the quality of the gears | Shared Source |
| Bending Strength of Gear Material | [170, 230] MPa | Gear manufacturing | Shared Source |
| Contact Strength of Gear Material | [590, 660] MPa | Machine Design (Norton 2000)and Materials Design (Ashby 1999) | Shared Source |
| Shock Factor | [1.25, 1.75] | Appropriate shock factor is not known for this application | Shared Source |
| Geometric Bending Stress Factor | [0.24, 0.49] | Dictated by the geometry of the gears – could be calculated but extremely difficult to do so | Not a shared source |
| Mass Variation | Equation Based – See Equation | Could be looked up for most gears but a model is used instead to apply to all gears | Not a shared source |

**Table 13. Formulation of Mini-Baja Gearbox Design Problem**

**Maximize**

Utility

$$U = U\left(a_{initial}\right)U\left(v_{\max}\right)U\left(r_{life}\right)U\left(m\right),$$

where $U\left(a_{initial}\right)$, $U\left(v_{\max}\right)$, $U\left(r_{life}\right)$, and $U\left(m\right)$ are the utility from the individual attributes. These are calculated by the following:

$$U\left(a_{\text{vehicle}}\right) = 1 - \exp\left(-\left(\frac{a_{\text{vehicle}}}{2}\right)^{0.75}\right),$$

where $U\left(a_{initial}\right)$ is the utility from the acceleration, $a_{initial}$ is the instantaneous acceleration at start-up, and are the shape and scale are factors in line with my preferences.

$$U\left(v_{\max}\right)=1-\exp\left(-\left(\frac{v_{\max}}{14}\right)^{1.375}\right),$$

where $U\left(v_{\max}\right)$ is the utility from the top speed, $v_{\max}$ is the top speed, and the shape and scale are factors in line with my preferences.

$$U\left(R_{\text{gearbox}}\right)=R_{\text{gearbox}},$$

where $U\left(r_{life}\right)$ is the utility from the reliability, $r_{life}$ is the reliability at 200 kilometers, and the shape and scale are factors in line with my preferences.

$$U\left(M_{\text{gearbox}}\right)=\exp\left(-\left(\frac{M_{\text{gearbox}}}{5}\right)^{0.5}\right),$$

where $U\left(m\right)$ is the utility from the mass, $m$ is the mass of the 3 gears, and the shape and scale are factors in line with my preferences.

**Select**

Gear Ratio ($N_g$)

$N_g=\left[1,5\right]$ (torque ratio)

Input Gear Diameter ($d_{in}$)

$d_{in}=\left[1.5,15\right]cm$

Idler Gear Diameter ($d_{id}$)

$d_{id}=\left[1.5,15\right]cm$

Gear Width ($w$)

$w=\left[1.27,\ 8.75\right]cm$

Gear Module ($p$)

$p=\left[1.27,\ 8.75\right]mm/tooth$

**Where**

Car Mass $M_{\text{car}}$

= [180, 205] kg. (~[400, 450] lbs)

Driver Mass $M_{\text{driver}}$

= [57, 102] kg. ([125, 225] lbs)

Engine Torque $T_{\text{engine}}$

$=B_1\omega_{\text{engine}}+B_2\omega_{\text{engine}}^2+B_3\omega_{\text{engine}}^3\pm\varepsilon_{\text{torque}}$

where $B_1=-7.82*10^{-3}\ N*m/rpm$, $B_2=-1.05*10^{-6}\ N*m/rpm^2$, $B_3=1.92*10^{-10}\ N*m/rpm^3$

Drag Coefficient*Area $C_{\text{drag}}$

= [0.197, 0.348] $n/\left(m/s\right)^2$

Coefficient of Internal Friction $C_{\text{internal}}$

= [0, 0.01471] $n/\left(m/s\right)$

Percentage of Mass on Rear Wheels $M_{\text{dist}}$
= [50, 75] %
Coefficient of Friction between Terrain and Wheels $\mu_{\text{wheels}}$
= [0.50, 1.00]
Gear Quality $Q_{\text{gears}}$
= [7, 12]
Gear Material Bending Strength $s_{\text{bending}}$
= [170, 230] MPa
Gear Material Contact Strength $s_{\text{contact}}$
= [590, 660] MPa
Gear Application Factor $K_{\text{A}}$
= [1.25, 1.75]
Gear Bending Stress Factor $J$
= [0.24, 0.49]
Tire diameter $d_{\text{wheels}}$
= 0.559m
Chain drive ratio $\rho_{\text{chain}}$
= 2.83
CVT ratio $\rho_{\text{CVT}}$
= [0.73, 3.83]
Engine Speed $\omega_{\text{engine}}$
= [1000, 3600] rpm

*Force produced by the drivetrain:*

$$F_{\text{wheels}} = \frac{2T_{\text{wheels}}}{d_{\text{wheels}}}$$

where $T_{\text{wheels}}$ is the torque transmitted from the engine:

$$T_{\text{wheels}} = \rho_{\text{CVT}}\rho_{\text{chain}}\rho_{\text{gearbox}}T_{\text{engine}}\left(\omega_{\text{engine}}\right)$$

where $\rho_{\text{gearbox}}$ is a design variable, $\rho_{\text{CVT}}$ is assumed to be 0.73, and $\omega_{\text{engine}}$ is related to the vehicle velocity:

$$\omega_{\text{engine}} = \frac{\rho_{\text{CVT}}\rho_{\text{chain}}\rho_{\text{gearbox}}V_{\text{vehicle}}}{2\pi r_{\text{wheels}}}$$

*Force of friction:*

$$F_{\text{friction}} = F_{\text{internal}} + F_{\text{air}}$$

where:

$$F_{\text{internal}} = C_{\text{internal}}V_{\text{vehicle}}$$

$F_{\text{air}} = C_{\text{drag}}V_{\text{vehicle}}^2$ and $C_{\text{drag}}$ is bound by the interval given above

*Maximum Velocity Calculation*

| |
|---|
| At this velocity the forces are balanced:<br>$F_{\text{friction}} = F_{\text{wheels}}$ |
| *Acceleration Calculation*<br>The instantaneous acceleration is calculated for $V_{\text{vehicle}} = 0\, m/s$ and $\omega_{\text{engine}} = 1000 rpm$, using the following equation:<br>$a_{\text{vehicle}} = \dfrac{\sum F}{m_{\text{vehicle}}}$<br>where:<br>$\sum F = F_{\text{friction}} + F_{\text{drivetrain}}$ |
| *Reliability Calculation*<br>The gearbox reliability is computed:<br>$R_{\text{gearbox}} = R_{\text{input}} R_{\text{idler}} R_{\text{output}}$<br>where $R_{\text{input}}$, $R_{\text{idler}}$, and $R_{\text{output}}$ are the reliability of the respective gears, computed from the following:<br>$R_{\text{gear}} = R_{\text{contact}} R_{\text{bend}}$<br>where both $R_{\text{contact}}$ and $R_{\text{bend}}$ are computed from:<br>$R = 1 - \exp\left(\left(\dfrac{-K_R}{0.699}\right)^{4.26}\right)$<br>where for $R$, $K_R$ is determined from:<br>$K_R = \dfrac{S}{\sigma}$<br>where $S$ is the bending strength and $\sigma$ is the bending stress for $R_{\text{bend}}$; $S$ is the contact strength and $\sigma$ is the contact stress for $R_{\text{contact}}$.<br>The bending stress is determined from:<br>$\sigma_b = \dfrac{F_t}{WmJ} \dfrac{K_a K_m}{K_v} K_s K_B K_I$<br>$F_t$ is the tangential force on the gear teeth; $W$ is the width of the gear; $m$ is the module; $J$ is the dimensionless geometry factor; $K_a$ is the application factor given as an uncertainty; $K_m$ is the load distribution factor, which is obtained from a look-up table from (Norton 2000); $K_s$ is the size factor determined from a table in , $K_B$ is the rim thickness factor that incorporates stressed caused by a large gear hub; $K_I$ is idler factor, which is 1.41 for the idler gear and 1 otherwise; $K_v$ is the dynamic factor determined from the following:<br>$K_v = \left(\dfrac{A}{A + \sqrt{200 V_t}}\right)$<br>where $V_t$ is the tangential velocity, and $A$ is given by:<br>$A = 50 - 56(1 - B)$ |

where $B$ is computed with:

$$B = \frac{\left(12 - Q_v\right)^{2/3}}{4}$$

where $Q_v$ is the quality of the gears, an uncertain parameter.

The tangential velocity is computed:

$$V_t = \frac{\omega_{\text{engine}}}{\rho_{\text{CVT}}} \frac{d_{\text{input}}}{2}$$

where $d_{\text{input}}$ is the diameter of the input gear, a design variable; $\omega_{\text{engine}}$ is the engine speed, which is assumed to be at the maximum torque speed of 2600rpm; $\rho_{\text{CVT}}$ is the ratio of the CVT, which is assumed to be at the maximum of 3.83.

The tangential force is calculated:

$$F_t = \frac{2\rho_{\text{CVT}} T_{\text{engine}}}{d_{\text{input}}}$$

where $T_{\text{engine}}$ is assumed to be at the maximum engine torque.

The bending strength is determined from the following:

$$S_{fb} = \frac{K_L}{K_T} S'_{fb}$$

where $K_T$ is the temperature factor, which equals 1 in this design; $K_L$ is the life factor, computed with the following:

$$K_L = 4.9404 N^{-0.1045}$$

when $N$, the number of cycles in a lifetime is below 3 million, and when the lifetime is above 3 million cycles the following is used:

$$K_L = 1.3558 N^{-0.0178}$$

The number of cycles in a lifetime is computed for the input gear:

$$N_{\text{input}} = \frac{l_{\text{life}}}{\pi d_{\text{wheels}}} \rho_{\text{chain}} \rho_{\text{gearbox}}$$

$\rho_{\text{CVT}}$ is the ratio of the CVT, which is assumed to be at the maximum of 3.83.

And for the idler gear:

$$N_{\text{idler}} = N_{\text{input}} \frac{d_{\text{input}}}{d_{\text{idler}}}$$

And for the output gear:

$$N_{\text{output}} = N_{\text{input}} \rho_{\text{gearbox}}$$

The contact strength is determined from the following:

$$S_{fc} = \frac{C_L C_H}{C_T} S'_{fc}$$

where $C_T$ is the temperature factor, which equals 1 in this design; $C_H$ is the hardness factor, which equals 1 in this design; $C_L$ is the life factor, determined from:

$$C_L = 2.466 N^{-0.056}$$

| |
|---|
| The contact stress is determined from:<br>$\sigma_c = C_p \sqrt{\frac{F_t}{WId}\frac{C_a C_m}{C_v} C_s C_f}$<br>$F_t$ is the tangential force on the gear teeth; $W$ is the width of the gear; $I$ is the dimensionless geometry factor; $C_a$ is the application factor, which is equal to $K_a$; $C_m$ is the load distribution factor, which is equal to $K_m$ and obtained from a look-up table from (Norton 2000); $C_s$ is the size factor, which is equal to $K_s$ and determined from a table in (Norton 2000); $C_f$ is the surface finish factor, which is 1 for this design; $C_v$ is the dynamic factor, which is equal to $K_v$. |
| *Mass Calculation*<br>Upper bound on gear mass<br>$M_{g,\text{upper}} = c_{\text{upper}} wd^2 + \varepsilon_{\text{upper}}$<br>where $c_{\text{upper}} = 0.0077\,\text{kg/cm}^3$, $\varepsilon_{\text{upper}} = 0.023\,\text{kg}$<br>Upper lower on gear mass<br>$M_{g,\text{lower}} = c_{\text{lower}} wd^2 + \varepsilon_{\text{lower}}$<br>where $c_{\text{lower}} = 0.0064\,\text{kg/cm}^3$, $\varepsilon_{\text{lower}} = -0.037\,\text{kg}$ |

With the basics of the problem formulated, attention is now turned to implementing a method for computing the bounds on performance in a manner consistent with the elimination principle presented in Chapter 4.

**Implementation: Computational Modeling in Matlab**

As described in Chapter 4, a design alterative should be eliminated if it is dominated by at least one other alternative. When the performance of the designs is formulated in terms of utility, this translates to the simple comparison:

$$\text{If } \overline{U}\left(A_j\right) < \max_{A_i \in D} \underline{U}\left(A_i\right) \quad \text{then eliminate alternative } A_j .$$

To make this comparison, the bounds of utility must be computed for the design alternatives. The implementation of this is the focus of this section.

To bound the performance of the system for the uncertain conditions, an approach is used that is similar to the one used in Monte Carlo Simulations (reviewed in Section 2.1.4). Specifically, the design artifacts are treated as deterministic black boxes with uncertain inputs, as shown in Figure 57, thus the only uncertainty experienced in the results is due to the uncertainty in the inputs to the model. The uncertainty from the model is included by attributing that uncertainty to uncertainty in the model parameters and then considering those parameters as inputs to the system. Using this approach, the inputs can be varied to determine the bounds on the performance that result from the model. I compute the bounds in this manner to avoid the dependence problem that can be easily encountered in complicated models with multiple instances of the same variable, as explained in Section 2.1.3. With this approach, a shared uncertain variable can be treated appropriately.

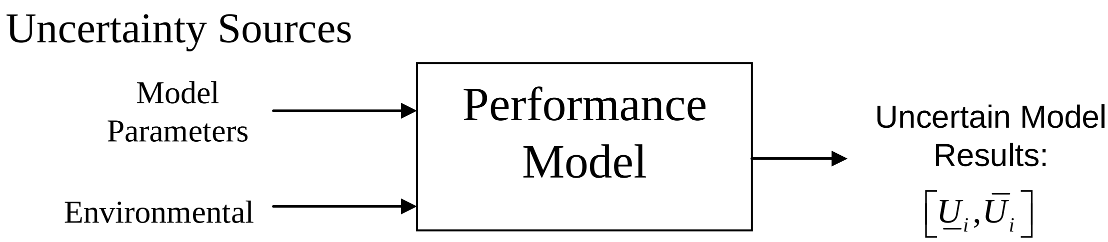



***Figure 57: The uncertainty is considered separate of the performance model for the designs.***

Unfortunately, computing the bounds on system performance can be computationally taxing. This is because the uncertain conditions that result in a bound are not known, so one must search through the uncertainty space to determine the bounds. This search can require significant resources, especially since this is required at each decision and for each alternative considered.

To ease the bounding process, an aspect of the performance can be exploited. If the performance function is known to be monotonic with respect to some of the uncertain variables then this can be exploited to simplify the computation of the bounds. This aspect was pointed out in Section 2.1.3 and Section 4.2, and it is employed here to help determine the bounds on performance. All of the uncertainties listed in Table 12 have a monotonic relationship with gearbox performance; for example, if an increase in the drag coefficient always results in a decrease in the performance of the vehicle then one knows that the maximum performance occurs at the minimum drag coefficient. Since this is the case for most of the uncertainty parameters that are considered, finding the bounds on performance is only a matter of determining whether the performance is increasing or decreasing with respect to the uncertain parameter. This is given in Table 14.

**Table 14: Relationship of uncertainty to gearbox performance**

| Source of Uncertainty | Performance Relationship to Increase in Uncertain Variable |
|---|---|
| Driver Mass | decreasing |
| Car Mass | decreasing |
| Engine Torque | increasing |
| Drag Coefficient * Area | decreasing |
| Coefficient of Internal Friction | decreasing |
| Percentage of Mass on Rear Wheels | increasing |
| Coefficient of Friction with Road | increasing |
| Gear Quality | increasing |
| Bending Strength of Gear Material | increasing |
| Contact Strength of Gear Material | increasing |
| Shock Factor | decreasing |
| Geometric Bending Stress Factor | decreasing |
| Cost Variation | decreasing |
| Mass Variation | decreasing |

To bound the performance in this manner, Matlab code is used. Specifically, 'utility_calc.m' is used to compute the deterministic performance of a design under specific conditions. Those conditions are supplied to the function by 'bounds_calc.m', which utilizes the built-in Matlab optimization function 'fmincon'. This function is used to search through the uncertainty to find the minimum and maximum performance for the given uncertainty and design. For this optimization function, the monotonic relationships pointed out in Table 14 are used to determine the starting points. Different starting points are used to check this monotonic nature.

While this process can produce the bounds for a specific design, it does not produce the bounds for a group of designs, which need to be considered in a B&B approach to design. For example, if one specifies a gear ratio for the gearbox, there are multiple designs available with that gear ratio and different input gear diameters, idler gear diameters, etc. These unspecified design variables are an additional source of uncertainty, as shown in Figure 58 that need to be included in the bounds. This uncertainty is not monotonic, so the bounds must be searched over the group of available designs. This is included in 'bounds_calc.m' by considering the unspecified design variables as more uncertain inputs to the system. This provides the bounds on the performance of a group of designs.

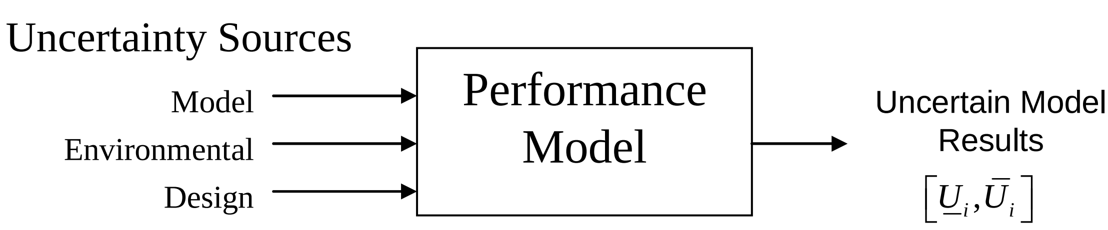



***Figure 58: The uncertainty is considered separate of the performance model for the designs. Included in this uncertainty is the unspecified design variables.***

While the process described enabled the bounds on performance to be calculated for a group of designs, this is not the best means of eliminating designs, as pointed out in Section 4.3. Instead, the relative performance of the designs needs to be considered, as it was in the elimination criterion:

$$\text{If} \max_{z_c \in Z_c, z_j \in Z_j, z_i \in Z_i} \left( U\left(A_j, z_j, z_c\right) - \max_{A_i \in D} U\left(A_i, z_i, z_c\right) \right) < 0 \text{, then eliminate } A_j .$$

To do so, the designs are compared with the same values of uncertainty that are shared, or common. This is displayed in Figure 59.

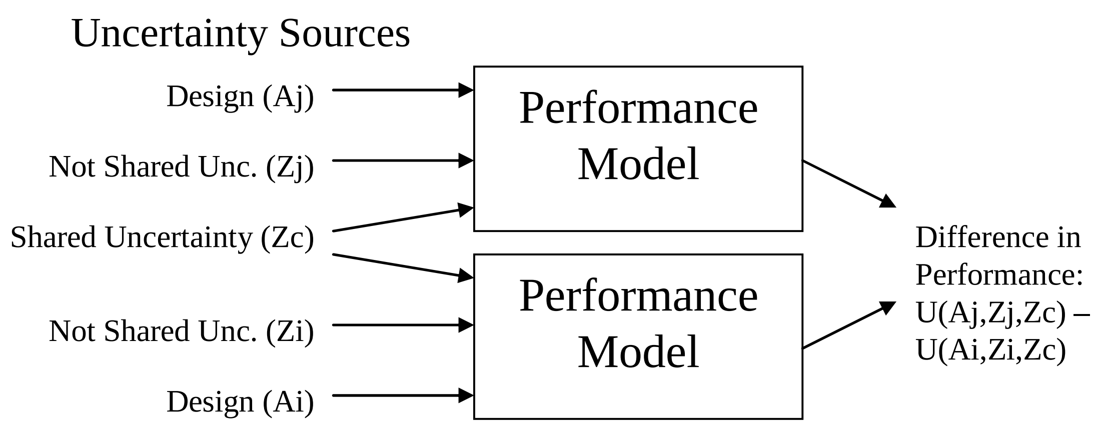



***Figure 59: The difference in performance is computed by considering the shared uncertainty the same for both performance models.***

To calculate the bounds on the difference in performance, the same search through the uncertainty space is used. However, this difference is not known to be monotonic with respect to the uncertain conditions. Therefore, the specific conditions at which the bounds occur are not known and a more extensive search is required. For this search, the ‘bounds_calc.m’ still employs ‘fmincon’ for computing the bounds, where the unspecified design variables are considered as uncertain parameters

To test the effectiveness of this elimination method, the design example is completed by eliminating as many designs as possible using the elimination method. The specifics of how this example is used to test the usefulness of the eliminate principle are discussed in the next subsection.

### 5.1.3 The Usefulness of the example design

In the previous subsection, the example design was explained. The purpose of the example is to test whether the elimination method is usefulness in engineering design. Specifically, the following concerns need to be tested in the example:

1. **Results in a reasonable number of designs to consider.** In eliminating designs, the objective is to reduce the number of designs considered. If this elimination principle cannot eliminate a significant number of designs then it could be costly for the B&B approach to converge or it could not converge at all. This ability to eliminate designs is limited by the uncertainty in the design – this is known. But how much uncertainty limits elimination is not known. If the uncertainty in a typical design seriously inhibits elimination, then this method may not be useful.
2. **The ability to converge through successive decisions.** While one needs to eliminate a significant number of designs in a decision to progress toward the solution, one also needs to do this in successive decisions to converge on the solution. It is possible that the

uncertainty in a typical design problem severely limits one's ability to make successive decisions, and one would converge only to a set of designs that is too large to be useful.

3. **Cost of applying the method.** For the method to be useful, it must not incur astronomical costs or require excessive time. In addition to the typical design costs, this includes costs associated with characterizing the uncertainty and computing bounds based on that uncertainty. This concern is considered with respect to the design alternatives eliminated. Incurring additional costs may be worthwhile if it results in significant additional elimination.

These issues are important in determining if the elimination method is useful; however, there are additional questions concerning usefulness of the Branch and Bound Approach to design. These questions require more examples in order to be answered:

**Is the method scalable?** The method is applied to the example problem of small scope, but it is not known whether the method can be effectively and efficiently applied to large systems and the involved limitations.

**How dependent is elimination on the branch (decision) chosen?** In branching, one determines the sets of designs considered for elimination; the sets that are determined effect which ones can be eliminated. How much the branching effects elimination is not known, and would need to be determined from multiple experiments.

These questions are considered in the example, but multiple examples are necessary to test these questions and determine an adequate answer. In this example the focus is on testing the usefulness of the elimination method; specifically, addressing the three issues relating to usefulness pointed out above. Since the gearbox example design tests all of these aspects, the example is a valid empirical example for testing the usefulness of the elimination method, fulfilling the second quadrant in the validation square. Attention is now focused on the design example and applying the elimination method in that example.

### *5.2 Elimination in the Example Design*

In this section, the gearbox example is used to test the elimination method. To do so, I proceed through the design process from a Branch and Bound Approach. Since this design method is not formalized, the steps of the process are determined as the design progresses. While this does not allow adequate testing of the Branch and Bound Approach, it does allow for testing of the elimination principle.

For the first elimination in the example design, the gear ratio is chosen. This selection is based on the intuition of the author. First, elimination is considered without using a reference design, and resulting bounds are shown in Figure 60. Based on these bounds, no elimination is possible. This is because the future design decisions could lead to a design that is an utter failure, as discussed in Chapter 4, and without making those decisions, one could not know that their outcome doesn't lead to failure. Additionally, all of the uncertainty in both designs is treated as unrelated, when in fact much of it is shared.

To combat these problems, a completed reference design is used to compare the relative performances possible with different gear ratios. Since the reference design is completed, there is no uncertainty about the other design variables; the variables have been specified and result in a usable gearbox design. The gearbox design from Ling and Bruns is used as a reference design

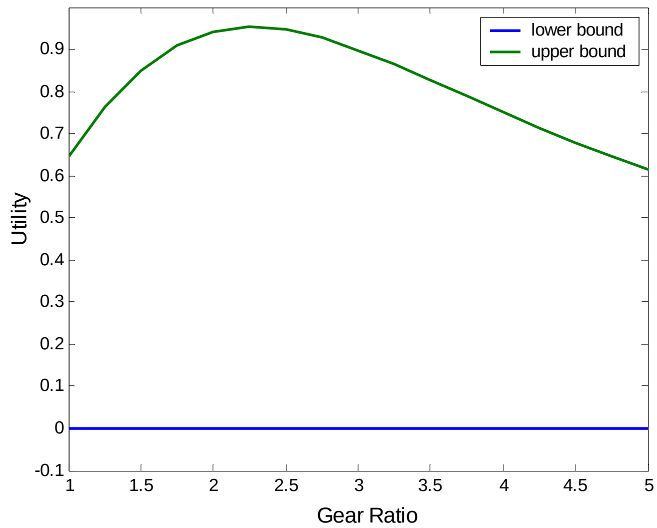



***Figure 60: Absolute utility bounds on the design performance based on gear ratio – no designs can be eliminated.***

(Ling and Brums 2004) and has the following specifications for the design variables: Gear ratio = 2.1, Input Gear Diameter = 1.5cm, Idler Gear Diameter = 1.5cm, Gear Width = 8.75cm, Gear Module = 6.35mm/tooth. They completed this design as part of a class project in optimization. Even though Ling and Bruns consider cost in their optimization, and the current design example does not, their design still serves as a good reference to begin the process.

Based on this reference design, the relative performance of the gearbox is computed again with respect to gear ratio. The results are shown in Figure 61. One notices that the upper bound on relative performance is only above zero between 1.9 and 3.7. This means only designs with these ratios could outperform the reference design; all other designs are dominated by the reference design and are eliminated. Thus, a significant number of designs can be eliminated when using this reference design that could not be eliminated otherwise.

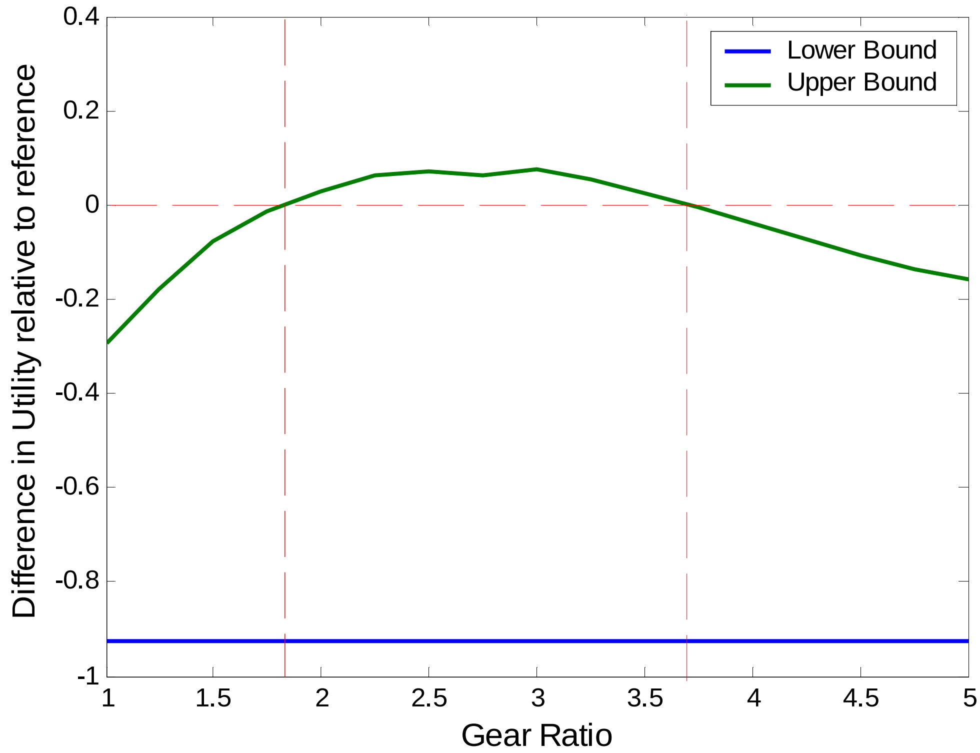



***Figure 61: Relative utility bounds based on the gear ratio compared to design provided by Ling and Bruns (Ling and Brums 2004)***

This elimination leaves the design tree as shown in Figure 62. As shown in the Figure, the entire remaining design space is considered as one branch, and this design space is next branched based on the input gear diameter. The absolute bounds and relative bounds with respect to the input gear diameter are shown in Figure 63. Once again, no elimination is possible by just considering the absolute utility bounds, but the upper bound on relative performance drops below zero after 6cm, allowing the elimination of all input gears greater than 6cm.

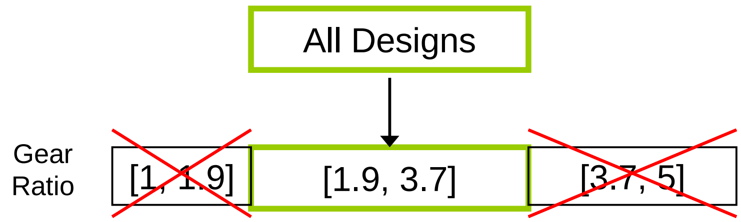



***Figure 62: Design branches after elimination based on gear ratio***

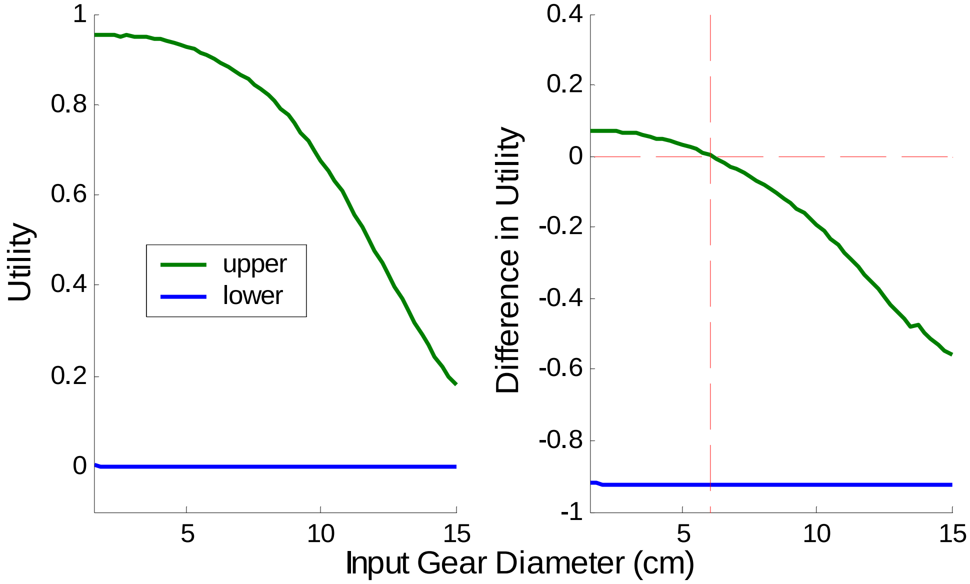



***Figure 63: Absolute bounds on the utility with respect to input gear diameter (left) do not allow any elimination, whereas relative bounds based do (right)***

The next branch in the design process considered for elimination is the gear width of all three gears. The absolute and relative bounds for this elimination are shown in Figure 5.5. Unfortunately, no designs can be eliminated with either the absolute bounds or the relative bounds. The uncertainty is too large, and the design variable's influence on performance is too weak to allow elimination with respect to this design variable. In the previous branches, the influence of the design variables on design performance is more clearly defined and much stronger. That is why elimination is possible at those branch levels, but not in this branch. Thus, the process moves on to the next branch with the understanding that a significant amount of the gear widths are still to be eliminated.

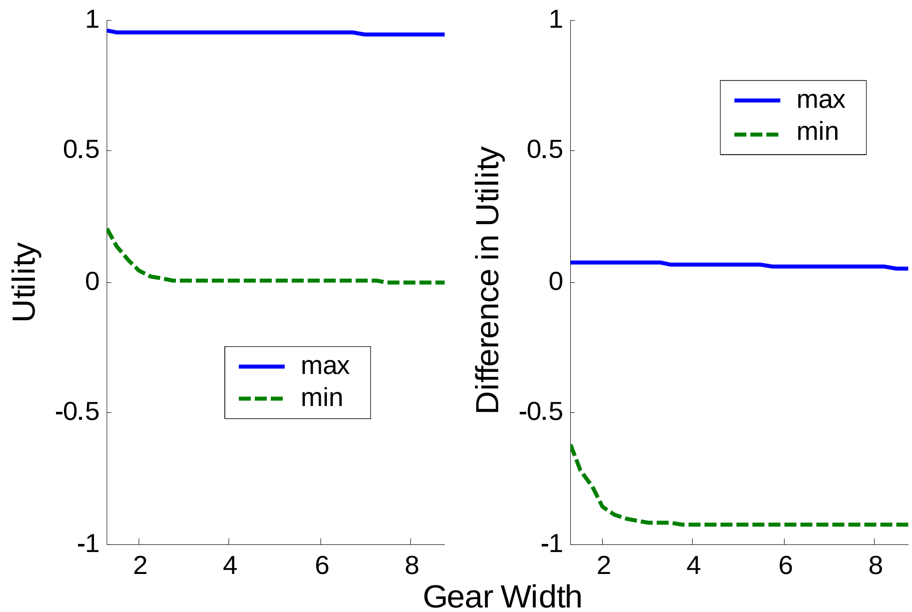



***Figure 64: Absolute (left) and relative (right) bounds on utility with respect to the gear width – no elimination is possible based on either set of bounds***

The next elimination branch is chosen as the idler gear diameter. The performance with respect to the idler gear diameter is shown in Figure 65. Once again, no elimination is possible, so the final design variable, the gear module, is considered in Figure 66. No elimination is possible

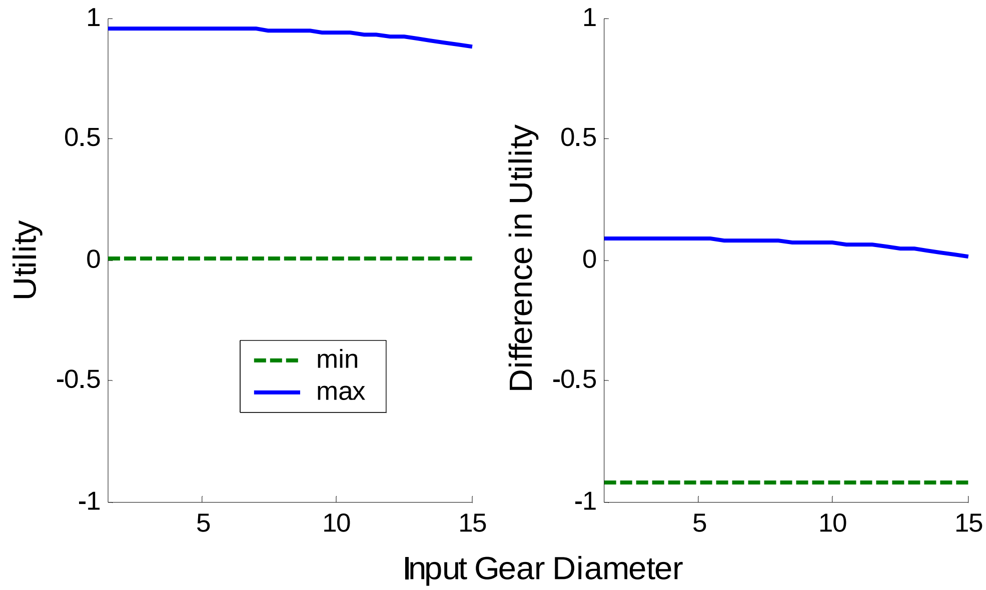



***Figure 65: Absolute (left) and relative (right) bounds on utility with respect to the input gear diameter – no elimination is possible based on either set of bounds***

No elimination is possible at the last three levels of the design branches, thus the design space remains as is shown in Figure 67. Even the use of reference design did not help eliminate designs, as it did in the first two branches. This is because the uncertainty from both the environment and the other design variables is too large and the impact of these design variables on the design is not as strong as for the gear ratio and input gear diameter. While this may be

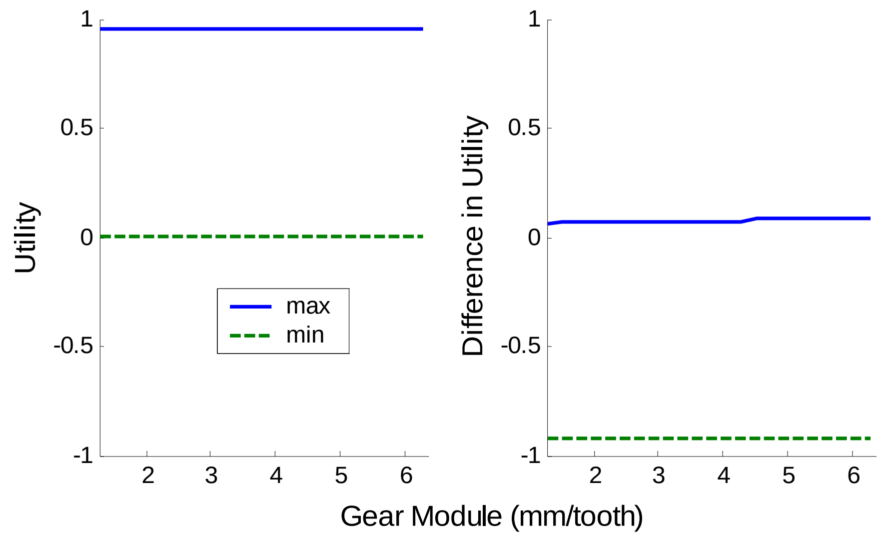



***Figure 66:   Absolute (left) and relative (right) bounds on utility with respect to the gear module – no elimination is possible based on either set of bounds***

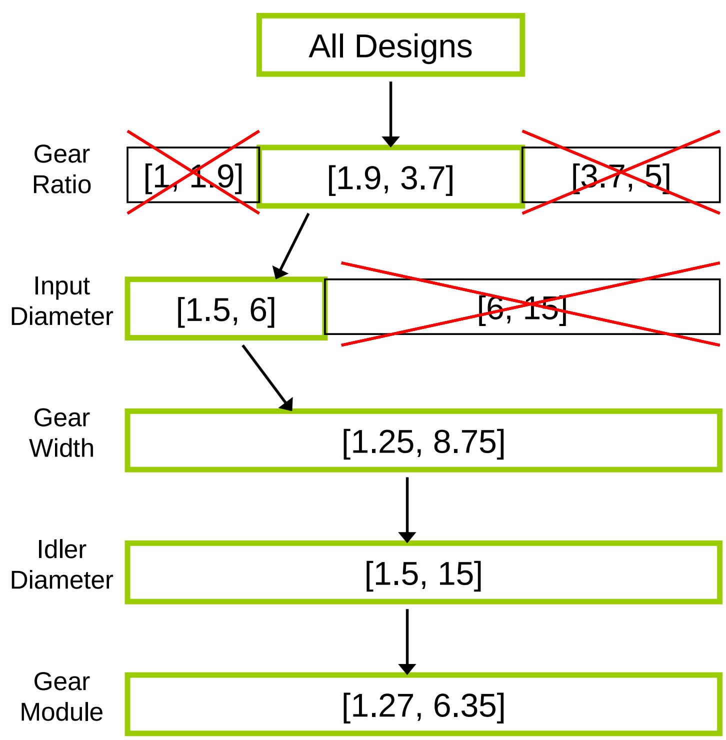



***Figure 67:   Design branches after first round of elimination.***

discouraging, more elimination is possible on the next iteration, as uncertainty about the other design variables is reduced, coupling between the branches is considered, and a different reference design is used.

In the previous iteration, no more elimination is possible with the reference design used, so in this iteration a new reference design is used for elimination. Using multiple reference designs is part of the method given in Chapter 4 and allows more elimination by taken advantage of the different impacts that shared uncertainty has on design performance. Based on the results of the previous iteration, I select the design variables for the reference design that appear most promising – that have the highest upper bound with respect to the previous reference design. Based on this process, I determined the new reference design: Gear ratio = 2.25, Input Gear Diameter = 1.5cm, Idler Gear Diameter = 1.5cm, Gear Width = 1.25cm, Gear Module = 6.35 mm/tooth. With this reference design, the process is iterated through the branches again.

The second iteration begins with the first branch: the gear ratio. The bounds on relative performance for the gear ratio compared to the new reference design are shown in Figure 68; based on the relative performance in this figure, the gear ratio greater than 3.0 are eliminated. At the next branch, the input gear diameter is compared to the reference again for elimination, as shown in Figure 69. With this maximum bound on relative performance, input gear diameters between 4.4 and 6 cm are eliminated. This leaves the design space as shown in Figure 70.

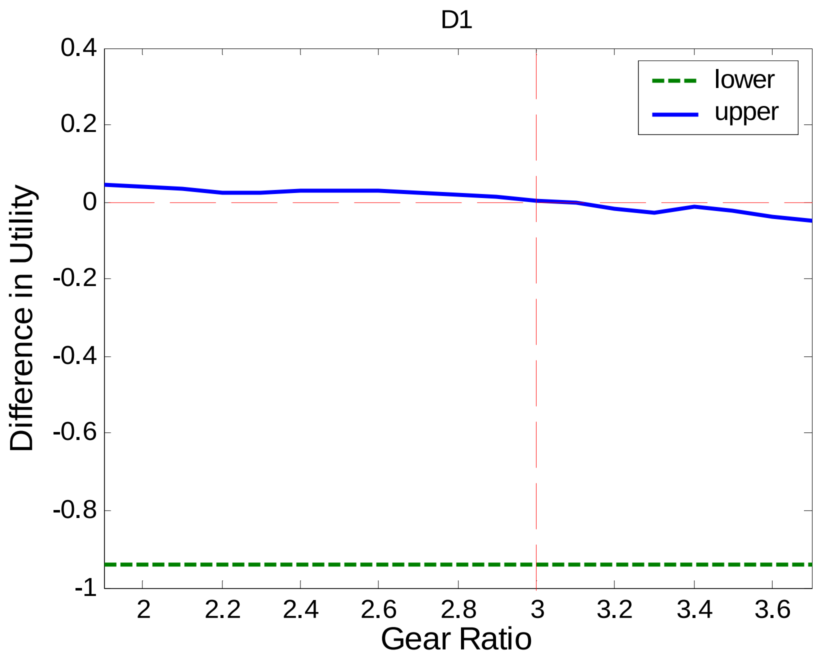



***Figure 68: Relative performance compared to new reference design and with respect to the gear ratio. Gear ratios greater than 3 are eliminated.***

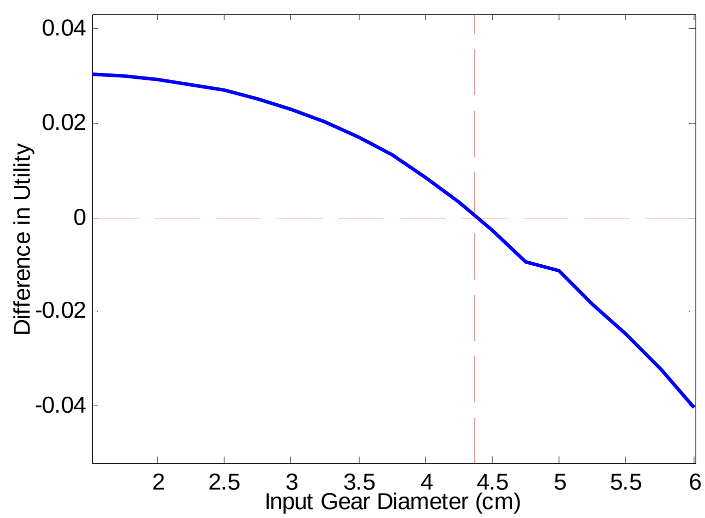



***Figure 69: Relative performance compared to new reference design and with respect to the input gear diameter. Input gear diameters greater than 4.4 are eliminated.***

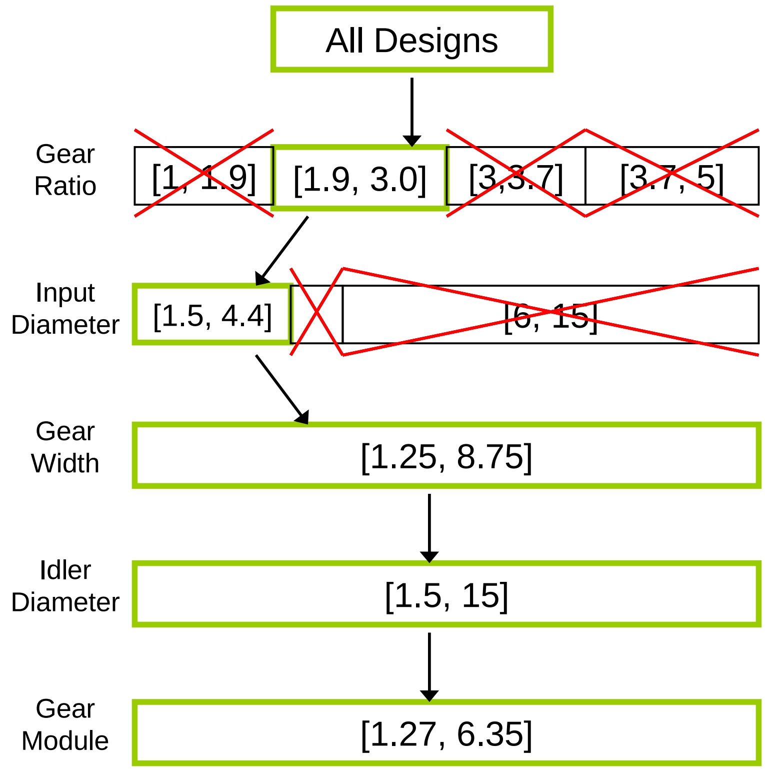



***Figure 70: Design branches after first two eliminations in the second iteration.***

In the previous iteration of the method, the next branches yielded no elimination. While, continuously iterating, with elimination in the gear ratio and input gear diameter, eventually may enable one to eliminate in the last three branches. Instead of taking this approach, I recognize that some of the design variables are strongly dependent on each other – they are coupled. To address this problem, elimination is no longer considered for only one design variable at a time. Instead two design variables are considered for elimination simultaneously. The difference in approach is contrasted in Figure 71Figure 70. In the diagram in the left of the figure, because there is significant coupling between the design variables and the elimination is only considered one design variable at a time, a significant number of inferior design alternatives are not eliminated. However, in the case on the right, where the two design variables are considered simultaneously, these inferior design alternatives can be eliminated. This demonstrates how considering strongly coupled design variables simultaneously allows more elimination.

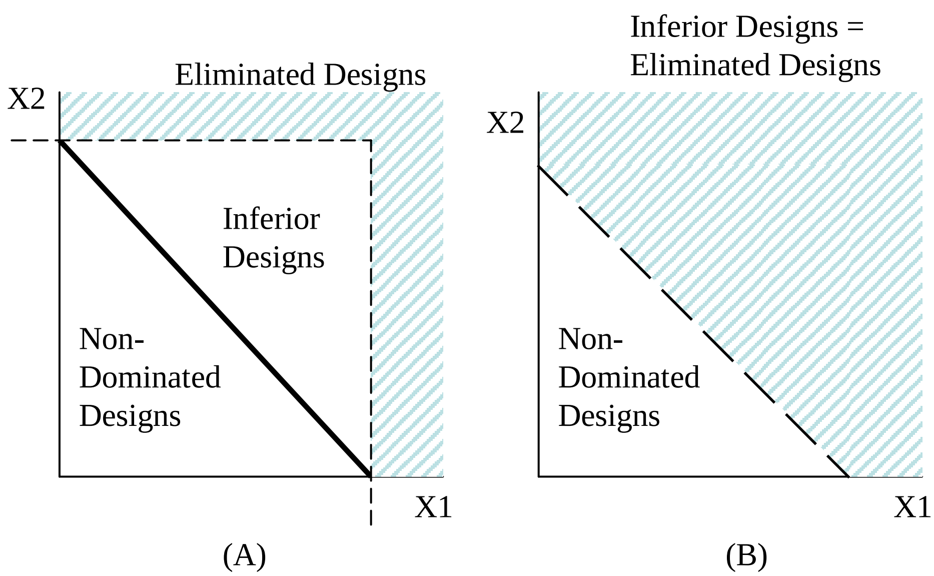



***Figure 71: Eliminating with respect to one design variable at a time leaves a significant number of inferior designs in the feasible set (left); however, these inferior design alternatives can be eliminated by simultaneously considering more than one design variable for elimination (right).***

To consider variables in tandem, the performance is evaluated with respect to specific values for both design variables based on the performance of the reference design. This is first applied to branches 3 and 4: gear width and idler gear diameter. The resulting upper bound relative to the reference is shown in Figure 72. Since the elimination criterion still applies, the design alternatives that have an upper bound on relative performance that is less than zero can be eliminated. These alternatives are distinguished in the plot on the right in Figure 72; the line, defined by $d_{\text{idler}} + 1.29w \le 10.9$, is used to divide the region of designs that is kept and that which is eliminated. As one can see from these plots, by considering two different branches simultaneously, more elimination is possible. This is due to the strong coupling between these two design variables.

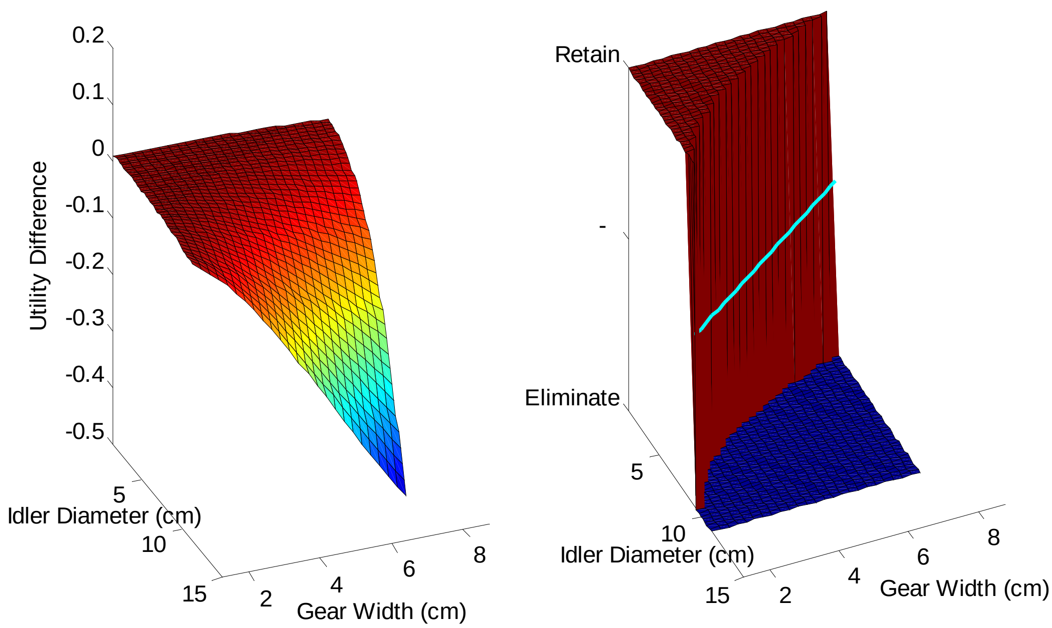



***Figure 72: By considering the relative performance of gear width and idler gear diameter at the same time in the plot on the left, a significant number of design alternatives can be eliminated, as shown in the plot on the right.***

Since this process is successful with branches 3 and 4, the same concept is applied to branches 2 and 4, in a hope to eliminate more designs. It is believed that these design variables are coupled strongly, as they are the two main factors in the mass of all three gears. In this situation, just as in the past coupled decision, a large idler gear diameter may be acceptable, but only in conjunction with a small input gear diameter. To apply this coupling in these decisions, the similar comparison is made: the designs are characterized for both input gear diameter and gear width. To allow for more elimination, the characterization is performed with respect to the new reference design used for this iteration. The result of these calculations is shown in Figure 73. Based on the bounds shown in Figure 73, the elimination is made, as shown in this figure as well. The blue line in the figure marks this elimination and is defined by $d_{\text{idler}} + 2.83 d_{\text{input}} \leq 14.9$. Considering these design variables simultaneously allows for elimination that was not possible otherwise.

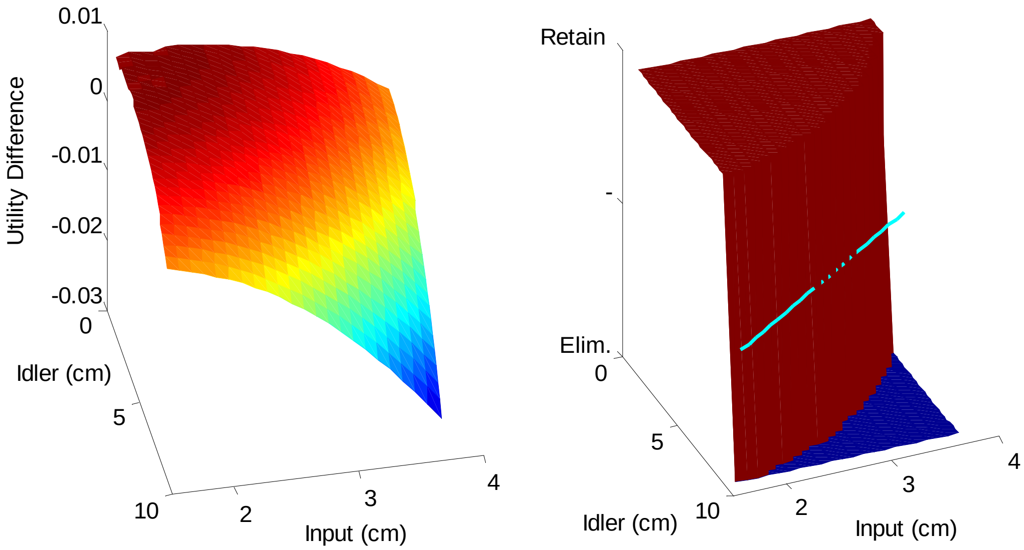



***Figure 73: The relative performance of the design alternatives are considered with respect to input gear diameter and idler gear diameter at the same time in the plot on the left, a significant number of design alternatives can be eliminated, as shown in the plot on the right.***

The final set of coupled design variables considered are the input gear diameter and the gear width, in branches 2 and 3, respectively. These design variables have a strong influence on the mass of both the input and output gears, as well as the reliability. The relative design performance is characterized with respect to these two design variables and is shown in Figure 74. As shown in this figure, the design space beyond the cyan line, defined by $d_{\text{input}} + 0.40w \leq 4.30$ can be eliminated. Thus, considering the coupling between these two design variables allows more elimination.

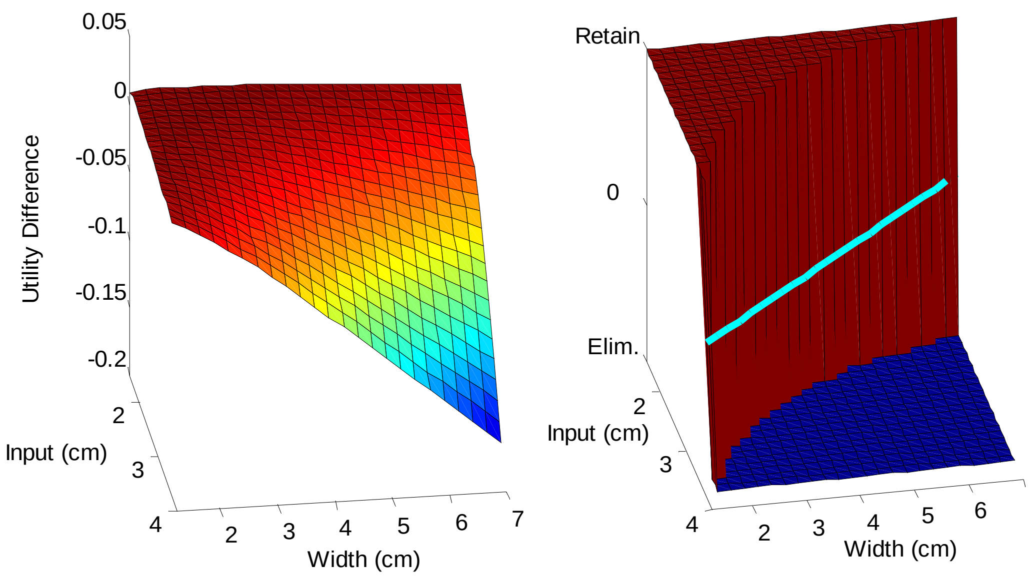



***Figure 74: Once again, by considering gear width and input gear diameter in the plot on the left, a significant number of design alternatives can be eliminated, as shown in the plot on the right.***

After this elimination, the design space remaining is not as easily defined in terms of branches. It is better defined in terms of constraints; these constraints are given in Table 5.1.

With this remaining design space and the environmental uncertainty, the performance of the resulting design in its environment could be between 0.52 and 0.95, and would vary from the reference design between -0.26 and 0.07. While this is significantly better than before elimination, this is still too large of a range to finalize the design

**Table 15: Design alternatives remaining after second iteration of elimination.**

| | |
|---|---|
| **Design Parameters:** | |
| Gear Ratio: | [1.9, 3] |
| Input Gear Diameter | [1.5, 3.7] cm |
| Idler Gear Diameter | [1.5, 9.25] cm |
| Gear Width | [1.25,7] cm |
| Gear Module | [1.27, 6.35] mm/tooth |
| **Additional Constraints:** | |
| Input and Idler Gear Diameters | $d_{\text{idler}} + 2.83 d_{\text{input}} \leq 14.9$ |
| Input Gear Diameter and Gear Width | $d_{\text{input}} + 0.40 w \leq 4.30$ |
| Idler Gear Diameter and Gear Width | $d_{\text{idler}} + 1.29 w \leq 10.9$ |
| **Utility of Design:** | |
| Upper Utility | 0.95 |
| Lower Utility | 0.52 |
| Upper Difference | 0.07 |
| Lower Difference | -0.26 |

Further elimination is not easy with the large uncertainty faced in the problem; this limitation was recognized in Chapter 3, where it was pointed out that a selection method may be necessary to complete the design process. Before discussing the results of this elimination, the design example is completed to show how one would go about completing the design from the resulting set of potential designs. In this case, there are three options that could be employed to complete the design. First, one could obtain more information that reduces the environmental uncertainty or model uncertainty. With less uncertainty in the problem, more designs can be eliminated. However, in the example design, this is not possible, as more precise information requires expensive equipment to obtain the data, such as a more expensive dynometer for a precise torque curve or a wind tunnel for a more precise drag coefficient. Thus, the uncertainty cannot be reduced within reasonable time and cost.

Second, one could continue the elimination by breaking the current space down into smaller subspaces. While this would be the best method for retaining the best design, it is not cost-effective. Computing bounds and keeping track of all of the resulting subspace is an extremely difficult without a proper method to do so. Additionally, I believe that with the uncertainty in this problem, the returns from proceeding in this manner would be minimal – the design space could not further be reduced. For these reasons, this approach is avoided.

Rather than to take these approaches, I focus on finding the most robust solution. To do so, I assume that the environmental and model uncertainties are the worst possible value for design performance. The design that performs best under these conditions is the most robust to adverse conditions. Even with these assumed conditions, the unspecified design variables are still a source of uncertainty, thus the elimination principle is still used to decide on the set of designs that has the most robust design. I continue the B&B design process by eliminating designs based on these deterministic parameters. Although this approach is used to finish the design, it is not a direct application of the method proposed in Chapter 4, where the design alternatives are evaluated considering all possible values for the uncertainty.

While, I have made the choice to select the designs based on the worst-case scenario of the uncertainty, this is not necessarily the preferable approach. One should consider the above options for their own situation and base their decision on the value they place on their resources and results. While the course of action in this situation is left for future research, attention is turned to finishing the design example.

## B&B Search using Deterministic, Worst-Case Information

For this remaining search, the B&B design approach is used to eliminate design alternatives without considering the environmental and model uncertainty. However, there is still uncertainty in the remaining problem. When the designer is eliminating based on one design variable, the other unspecified design variables are uncertain and limits elimination. To counter this, design alternatives still are compared to a reference design for elimination. The same reference design from the last iteration is used again: Gear Ratio = 2.1, Input Gear Diameter = 1.5 cm, Idler Gear Diameter = 1.5 cm, Module = 6.35 mm/tooth, Width = 1.25 cm. The first iteration through the branches is shown in Figure 75, as well as the design space that results from the iteration. This iteration moves closer to the final design, but the designs available still vary in utility between 0.64 and 0.95, and from the reference design between -0.10 and 0.06. Additionally, the idler gear diameter, the gear width, and gear module need much tighter ranges before the design is completed, so another iteration is performed. This iteration is shown in Figure 76 along with the constraints that define the remaining design space. This design space contains designs that vary in utility between 0.68 and 0.95, and differ from the reference design between -0.055 and 0.058. This variation from the reference design is small enough that the design can be finalized deterministically.

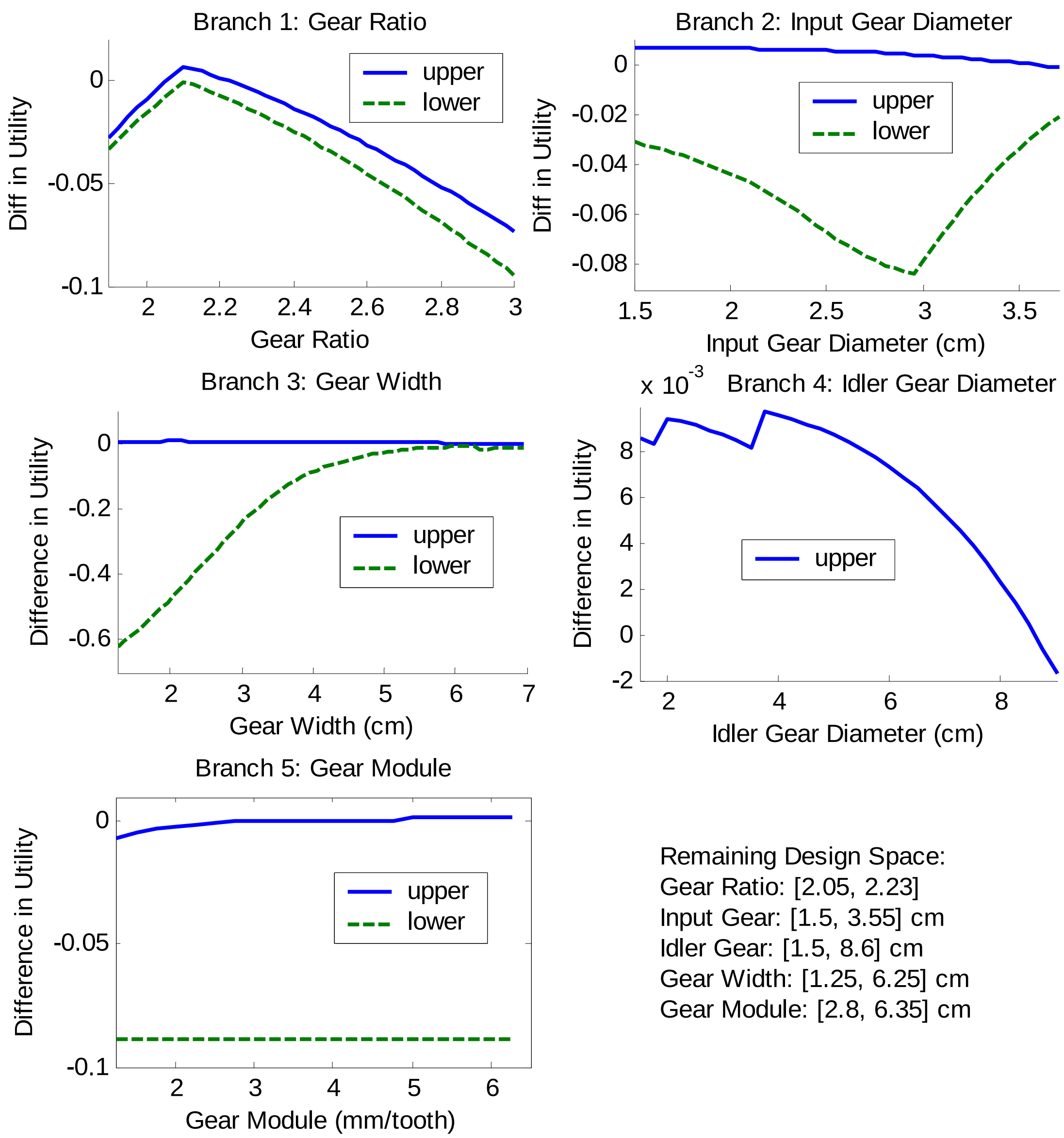



***Figure 75: Elimination in the first iteration of the deterministic B&B search based on the worst-case.***

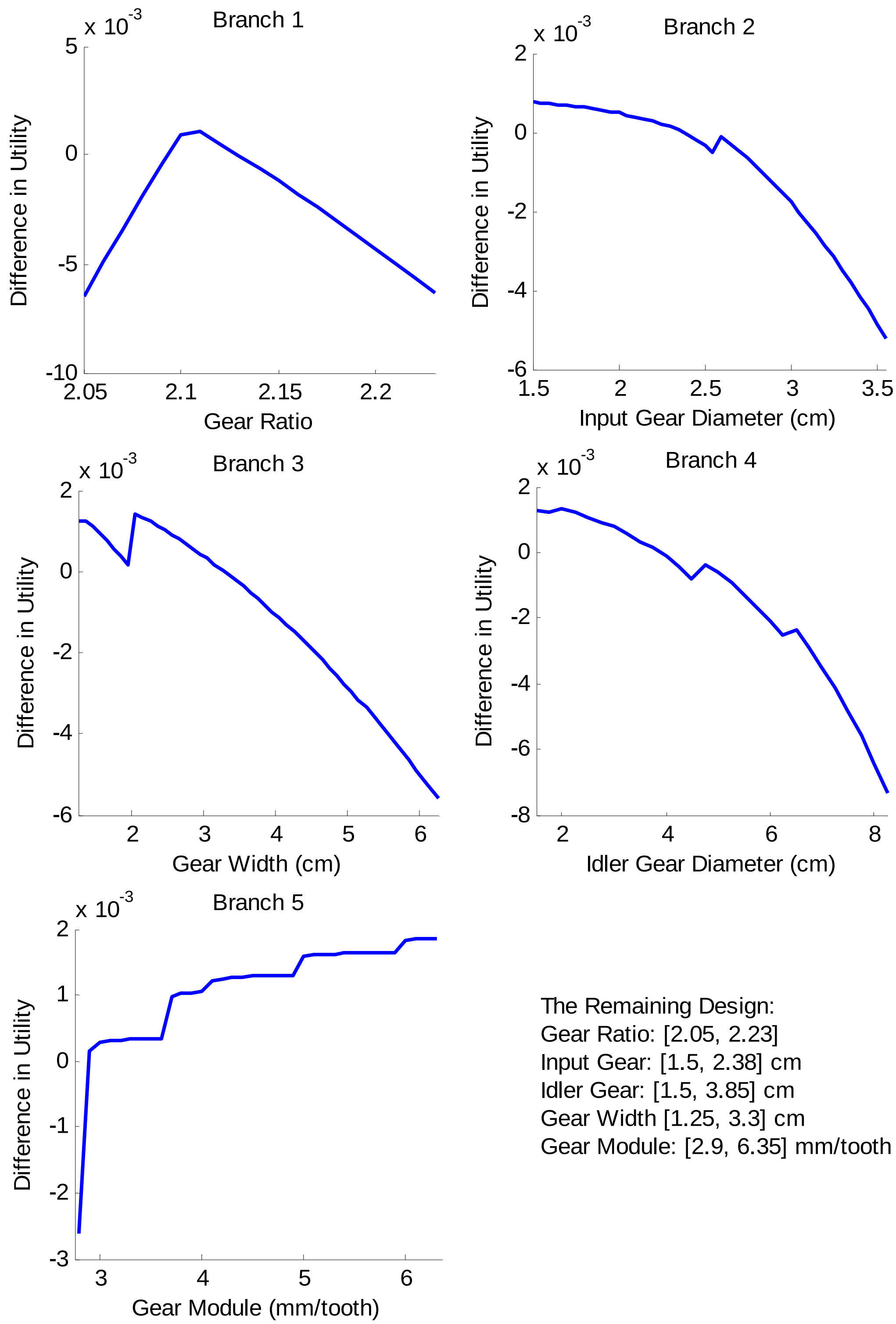



***Figure 76: Elimination in the second iteration of the deterministic B&B search based on the worst-case.***

To close in on a specific design from the remaining space, the most robust designs are chosen based on the selecting the maximum lower bounds. This leads to the following design given in Table 16.

**Table 16: Design resulting from deterministic B&B search based on worst-case uncertainty**

| | |
|---|---|
| **Design Parameters:** | |
| Gear Ratio: | 2.1 |
| Input Gear Diameter | 1.5 cm |
| Idler Gear Diameter | 1.5 cm |
| Gear Width | 1.25 cm |
| Gear Module | 6.35 mm/tooth |
| **Utility of Design:** | |
| Upper Utility | 0.95 |
| Lower Utility | 0.73 |

Unfortunately, this does not work out to actual gears that can be manufactured. This demonstrates how the design variables one selects need to be appropriate for the practical design problem. To consider the practical aspects of the problem, adjustments are made to the gear module to result in the design:

**Table 17: Design resulting incorporating the practical considerations of the problem based on the results from deterministic B&B search based on worst-case uncertainty**

| | |
|---|---|
| **Design Parameters:** | |
| Gear Ratio: | 2.1 |
| Input Gear Diameter | 1.5 cm |
| Idler Gear Diameter | 1.5 cm |
| Gear Width | 1.25 cm |
| Gear Module | 1.50 mm/tooth |
| **Utility of Design:** | |
| Upper Utility | 0.95 |
| Lower Utility | 0.73 |
| **Physical Design:** | |
| Input Teeth: | 10 |
| Idler Teeth: | 10 |

| | |
|---|---|
| Output Teeth: | 21 |

While this is not the design determined from the deterministic search, it is within the range of designs that resulted from the application of the elimination method under all the uncertain conditions. Using the elimination method, this design was still considered, it is only afterward that it no longer was considered.

While this design example demonstrates how the elimination method is useful, in the design process, this does not demonstrate how each aspects of the method aids elimination, nor is this example compared with other approaches to design. These issues are addressed in the next section, where the design example is analyzed.

## *5.3 Analysis of Example Results*

In this section the design example is analyzed to demonstrate that the elimination method is useful and each aspect of the method is useful. First, the design produced with the B&B approach is compared to designs produced with two different deterministic approaches. Then the use of a reference design in the elimination method is investigated; I demonstrated how using a specified reference design aids elimination by taking advantage of shared uncertainty and removing uncertainty about the other design variables.

### 5.3.1 Comparison to Deterministic Approaches

The design example in the previous section demonstrates how the elimination method can be used to significantly reduce the design alternatives considered, and thus is useful in design. However, one does not know how this approach compares to the typical approaches to design. To assess this, the resulting design is compared with two designs resulting from point-and-iterate deterministic approaches. The B&B produced design should at least be comparable, and hopefully superior, to the designs produced with the deterministic approaches.

In the first design obtained by the deterministic approach, all the uncertain parameters are assumed to be their median value, or mid-interval. For example, the bending stress of the gear material is [170, 230] MPa, and is assumed to be 200MPa. In a decision, these assumed values are used to compute the performance of the design with respect to the design variable in that decision. The value for the design variable that produces the design with the highest utility is selected. This continues for each decision in the example. The process continues iterating through the same decisions until the utility between iterations is less then the designer's indifference to iterating, specified in this case to a utility difference of 0.01. This design process is diagramed in Figure 77. While this defines the starting point of the algorithm, this point-and-iterate approach needs a starting point from which to iterate; for this the design provided by Ling and Bruns is used (Ling and Brums 2004). This has the following variable values: Gear Ratio = 2.1, Input Gear Diameter = 1.5 cm, Idler Gear Diameter = 1.5 cm, Module = 1.27 mm/tooth, Width = 8.75 cm.

In the second design from a deterministic approach, the uncertain parameters are assumed to be the worst-case possible values. For example, the bending stress of the gear material, which falls within [170, 230] MPa, is assumed to be 170MPa. These values are used with the same point and iterate approach, the same starting point, and continue until the difference in utility between iterations is less than the 0.01 indifference.

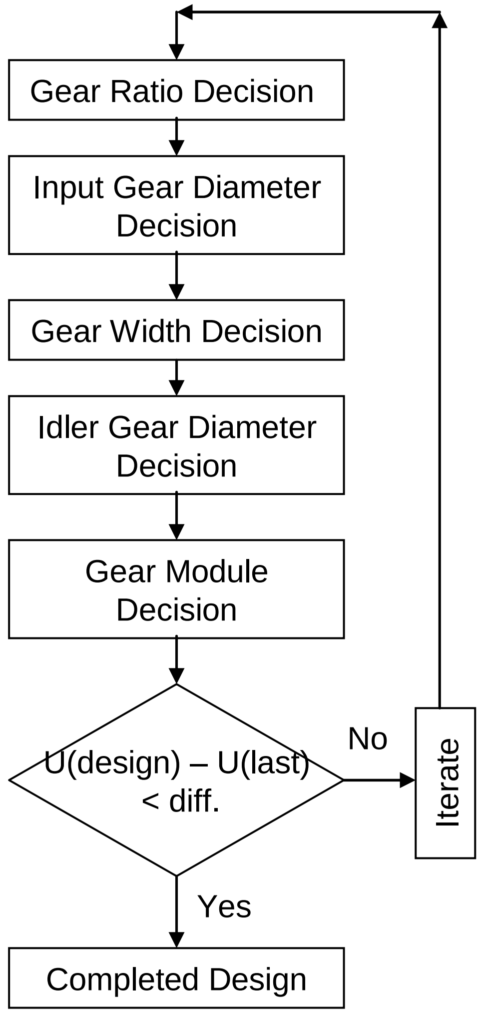



***Figure 77: Point-and-iterate process used in deterministic search for gearbox design***

The results of the two deterministic approaches are shown in Table 18, along with the design from the proposed B&B approach. As one can see from the table, the design obtained from the B&B approach, using the elimination method, is very similar in design and performance to the design obtained through the deterministic approaches. To show the similarities, the bounds on the utility for each design are shown in Figure 78. As seen in the figure, the design resulting from the B&B approach can only be differentiated from the two other designs in close-ups on the bounds. While this demonstrates that the elimination method and B&B approach can result in a design that is as good, if not better, than as those obtained through the traditional point-and-iterate methods, these results also demonstrate that the B&B approach may not be necessary to find a good design. Since the design obtained by the B&B approach differs little from those obtained deterministically and the B&B approach is more costly, using the B&B approach in this situation may not be the best decision economically.

**Table 18: Design from deterministic point-and-iterate searches and the elimination approach**

| | Median | Worst-Case | B&B method |
|---|---|---|---|
| Gear Ratio: | 2.4 | 2.1 | 2.1 |
| Input Gear Diameter | 1.5 cm | 1.5 cm | 1.5 cm |
| Idler Gear Diameter | 1.5 cm | 1.5 cm | 1.5 cm |
| Gear Width | 3.5 cm | 4.15 cm | 1.25 cm |
| Gear Module | 3.17mm/tooth | 3.17 mm/tooth | 6.35 mm/tooth |
| Upper Utility | 0.950 | 0.948 | 0.950 |
| Lower Utility | 0.712 | 0.732 | 0.733 |
| Iterations | 2 | 2 | 2 |

While this particular design can be completed more cost-effectively with a point-and-iterate approach, the B&B approach offers some significant advantages over these point-based approaches in other design problems. In design problems where the objective function is not as smooth then using the point-and-iterate based approaches could easily get one caught in a local maximum. This is especially true for complicated design problems that involve decisions on alternatives in multiple decisions. In these situations, a point-and-iterate method can result in a quick design, but the design may be substantially worse than the most-preferred design. In those

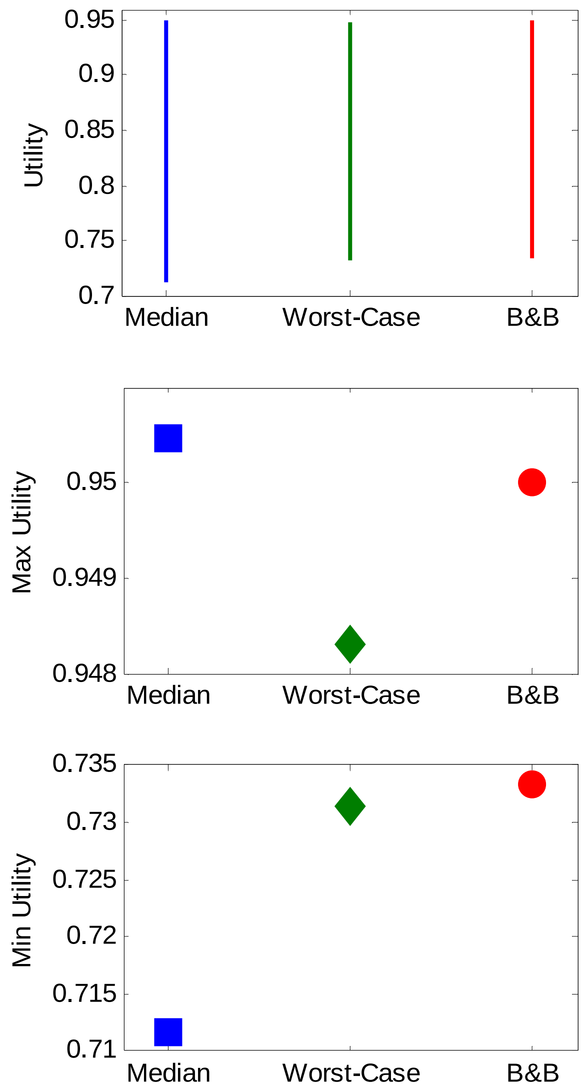



***Figure 78: Bounds of the two deterministic approaches versus the B&B elimination approach***

situations, the B&B approach with the proposed elimination could perform far superior to a point-and-iterate approach. The elimination approach would consider all the possible designs and not eliminate any of the designs that are potentially the most-preferred. Future examples that test out the effectiveness of B&B approach should include more complicated problems in which the utility function may have multiple local maximum.

Even in designs similar to the current problem, the B&B approach is useful. When one begins the design problem, one may not be aware of whether the design involves multiple local maximum in the utility function. Therefore, one can use the elimination method proposed to take the more robust approach to design. The value in converging on the most-preferred design is significant enough to warrant the extra expenses that can be incurred in doing so. Because of this robust quality of the method, it is valuable in design.

**Usefulness in converging on the most-preferred design**

While the previous comparison points out how the elimination method can be used to determine an actual design, this is not the purpose of the elimination method. The method is supposed to result in a set of designs that bound the most preferred design for all the conditions possible in the uncertainty. That is, the resulting set needs to contain designs that are most preferred for whatever environmental conditions result. Thus, to more accurately assess the value of the elimination method, one should look at the bounds that resulted from the process, and if the most preferred designs (for any set of conditions) falls within those final constraints. To make this comparison, the absolute best designs for each set of conditions are shown in Table 19 along with the final constraints that resulted from the elimination method. The designs selected for the specific conditions fit within the set of final designs resulting from the elimination process. This demonstrates that the elimination can be used to eliminate inferior designs without eliminating the most preferred designs. Thus, the method has proven to be robust for this design example.

**Table 19: The resulting set of designs from the B&B Design contains the designs most-preferred design under the worst-case conditions and the median uncertainty conditions.**

| | B&B method | Worst-Case | Median |
|---|---|---|---|
| Gear Ratio: | [2.1, 3] | 2.10 | 2.29 |
| Input Gear Diameter | [1.5, 3.7]cm | 1.5 cm | 1.5 cm |
| Idler Gear Diameter | [1.5, 9.25]cm | 1.5 cm | 1.5 cm |
| Gear Width | [1.25, 7]cm | 2.00 cm | 1.77 cm |
| Gear Module | [1.27, 6.35] mm/tooth | 6.35 mm/tooth | 6.13 mm/tooth |
| Additional Constraints | $d_{\text{idler}} + 2.83 d_{\text{input}} \leq 14.9$ | | |
| | $d_{\text{input}} + 0.40 w \leq 4.30$ | | |
| | $d_{\text{idler}} + 1.29 w \leq 10.9$ | | |
| Upper Utility | 0.955 | 0.950 | 0.954 |
| Utility (median unc) | - | - | 0.874 |
| Lower Utility | 0.52 | 0.734 | 0.723 |
| Iterations | 2 | 2 | 2 |

### 5.3.2 Comparison of each aspect of the method

While the previous section has shown that the elimination approach can close in on the best design for the variety of uncertain conditions, this does not show how each of the elements in the method has helped to do so. In this section, the different aspects that help eliminate the designs are investigated. Specifically, I pointed out how shared uncertainty, and considering a specific design as the reference for that shared uncertainty, allow one to eliminate more designs.

**Usefulness of Shared Uncertainty**

In the first branch of the design (gear ratio), designs with a gear ratio of greater than 3.0 were eliminate with the reference design. From Figure 60, Figure 61, and Figure 68 it should be apparent that a reference design is needed for elimination of these designs. However, one may just think that this reference allowed elimination because it specified other design variables that had not been fixed yet. This is not the case. The uncertainty shared between the designs is significant and needs to be considered to make this elimination possible. To show this, I compare two designs with the same future design variables but with gear ratios of 2.1 and 3.5. These designs are as given in Table 20. In this table, the absolute bounds are shown, as well as the relative bounds. While the absolute bounds show that the design with a gear ratio of 2.1 may be more promising and more robust, elimination is not possible based on these bounds. However, if one considers the relative bounds, the design with a gear ratio of 3.5 can be eliminated. This comparison of the bounds is illustrated in Figure 79.

**Table 20. Comparison of two designs differing only in gear ratio with and without shared uncertainty considered**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | 2.1 | 3.5 |
| Input Gear Diameter | 1.5 cm | 1.5 cm |
| Idler Gear Diameter | 1.5 cm | 1.5 cm |
| Gear Width | 1.25 cm | 1.25 cm |
| Gear Module | 6.35 mm/tooth | 6.35 mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.827 |
| Minimum Utility | 0.733 | 0.595 |
| Maximum Difference in Utility (B – A) | -0.022 | |
| Minimum Difference in Utility (B – A) | -0.167 | |

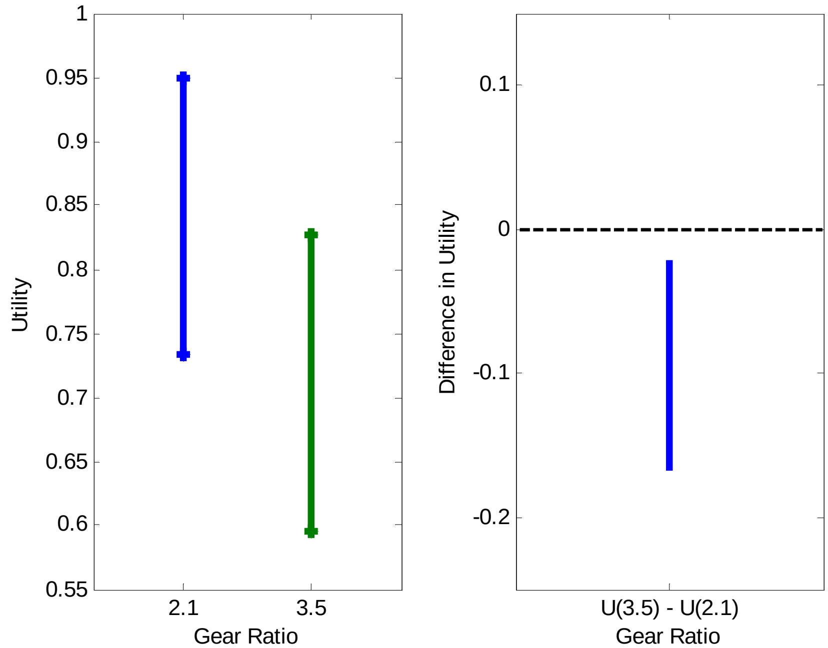



***Figure 79: The two design alternatives are compared based on absolute bounds in the figure on the left, and no elimination is possible; however, if common uncertainty is considered by computing the relative performance of the designs then Design A, with a gear ratio of 2.1 dominates Design B.***

To demonstrate further how shared uncertainty aides elimination, the same comparison for elimination is made for the input gear diameter. In this branch, the designs with an input gear diameter greater than 6 were eliminated. To show how shared uncertainty played a part in the process, the designs with input gear diameters of 1.5 cm and 7 cm and all other design variables the same, are compared. These designs, and their bounds, are given in Table 21, while the bounds alone are shown in Figure 80. Once again, elimination is not possible based on the absolute bounds, but Design B is dominated by Design A and can be eliminated if the shared uncertainty is considered by comparing the relative bounds. Through the two examples presented it should be clear that shared uncertainty significantly aids eliminating and should be considered when comparing relative performance.

**Table 21. Comparison of two designs differing only in input gear diameter with and without shared uncertainty considered**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | 2.1 | 2.1 |
| Input Gear Diameter | 1.5 cm | 7 cm |
| Idler Gear Diameter | 1.5 cm | 1.5 cm |
| Gear Width | 1.25 cm | 1.25 cm |
| Gear Module | 6.35 mm/tooth | 6.35 mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.857 |
| Minimum Utility | 0.733 | 0.643 |
| Maximum Difference in Utility (B – A) | -0.040 | |
| Minimum Difference in Utility (B – A) | -0.1156 | |

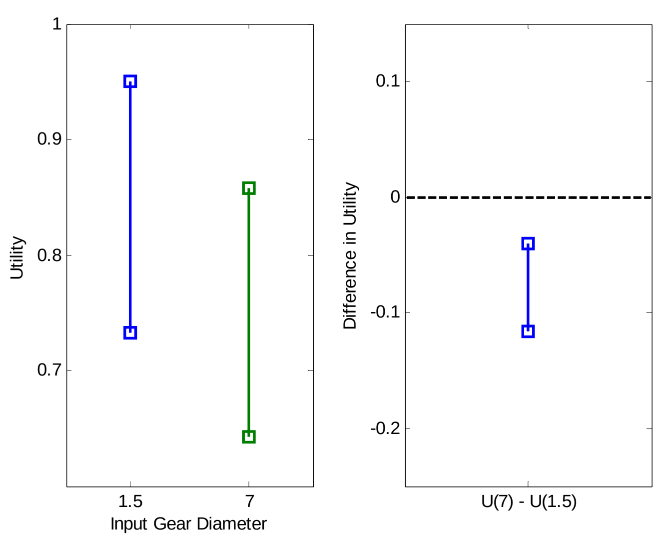



***Figure 80: The absolute bounds on the performance of each design allows no elimination, as shown in the plot on the left; however, if the relative performance of the designs is considered, then Design B, with a input gear diameter of 7cm can be eliminated, as displayed in the plot on the right***

## Usefulness of a Specific Reference Design

The previous examples shows how a reference design allows one to eliminate more by taking advantage of the shared uncertainty in the designs. A reference design can also reduce the uncertainty associated with unspecified design variables and allow more elimination, but to do so a specific reference design should be used. To demonstrate the need for a specific reference design, elimination is performed with a reference design that is just specified by the design variable currently considered for elimination as well as with a reference design that is fully specified.

In the design process, the gear ratio of 3.5 was eliminated using a reference design with a gear ratio of 2.1. These gear ratios are compared first without any of the other design variables specified, so the designs compared for elimination are as given in Table 22. The bounds for these sets of designs are given in the table and in Figure 81. Based on these bounds, no elimination is possible. Without the other design variables specified in the reference design, one cannot be sure that the other variables can be selected in a manner that produces a viable design. Thus, the absolute bounds on the reference design's utility include zero. The relative comparison performance includes this possibility, which results in large bounds on the difference in performance and no elimination.

**Table 22. Comparison of two designs differing only in gear ratio without any other design variables specified**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | 2.1 | 3.5 |
| Input Gear Diameter | [1.5, 15] cm | [1.5, 15] cm |
| Idler Gear Diameter | [1.5, 15] cm | [1.5, 15] cm |
| Gear Width | [1.25, 8.75] cm | [1.25, 8.75] cm |
| Gear Module | [1.27, 6.35] mm/tooth | [1.27, 6.35] mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.8268 |

| | | |
|---|---|---|
| Minimum Utility | 0.00 | 0.00 |
| Maximum Difference in Utility (B – A) | 0.7485 | |
| Minimum Difference in Utility (B – A) | -0.9375 | |

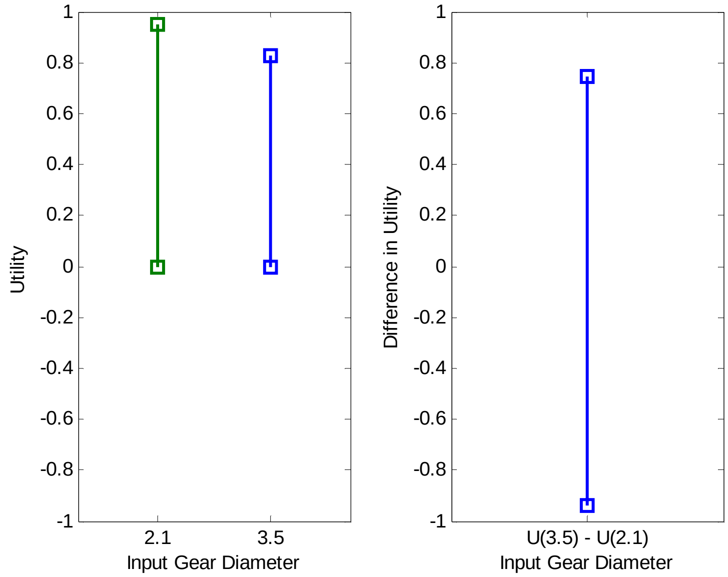



***Figure 81: Bounds of the two designs with respect to gear ratio and without any of the other design variables specified.***

To counter this problem, and reduce the uncertainty caused by these unspecified other design variables, one should use a specific design instance as a reference. For this example, the reference design is specified as given in Table 23. Because this reference design is fully specified, there is no uncertainty about its other design variables, thus there are much tighter bounds on this design than with the other design variables unspecified, as seen in Figure 82. This results in tighter bounds on the relative performance as well. Even with that reduction, the bounds on relative performance are still large because of the uncertainty in the Design B, but most importantly the relative performance does not include zero within the bounds. Thus, all designs with a gear ratio of 3.5 can be eliminated; this elimination is now possible because of the fully specified reference design.

**Table 23. Comparison of two designs differing in gear ratio; the reference design is fully specified**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | 2.1 | 3.5 |
| Input Gear Diameter | 1.5 cm | [1.5, 15] cm |
| Idler Gear Diameter | 1.5 cm | [1.5, 15] cm |
| Gear Width | 1.25 cm | [1.25, 8.75] cm |
| Gear Module | 6.35 mm/tooth | [1.27, 6.35] mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.8268 |
| Minimum Utility | 0.733 | 0.00 |
| Maximum Difference in Utility (B – A) | -0.0206 | |
| Minimum Difference in Utility (B – A) | -0.9375 | |

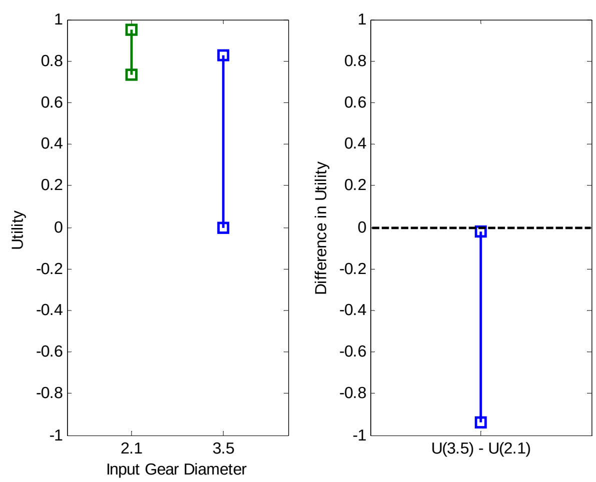



***Figure 82: Bounds of the two designs with respect to gear ratio with a fully specified reference design.***

To further demonstrate how this approach is necessary, the second branch is used to compare the elimination of the input gear diameter of 7 by the input gear diameter of 1.5cm. The designs compared are as given in Table 24 along with their performance bounds. Because the designs are not specified, the performance can fall within a broad range; this makes elimination impossible. So, instead of using this unspecified reference design, a specific reference design, given in Table 25 is used, and with this reference design, the bounds given in this table were obtained. Once again, these bounds are much tighter and allow elimination because a specific design is used. Using a specific reference design is an important part of the method that helps eliminate uncertainty due to other design variables. This importance is illustrated in Figure 83, where the bounds obtained with a specific reference design, in the bottom two figures, is much tighter those bounds obtained without a specified reference design, in the top two figures.

**Table 24. Comparison of two designs differing in input gear diameter without the other design variables specified.**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | [1.9, 3] | [1.9, 3] |
| Input Gear Diameter | 1.5 cm | 7 cm |
| Idler Gear Diameter | [1.5, 15] cm | [1.5, 15] cm |
| Gear Width | [1.25, 8.75] cm | [1.25, 8.75] cm |
| Gear Module | [1.27, 6.35] mm/tooth | [1.27, 6.35] mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.866 |
| Minimum Utility | 0.00 | 0.00 |
| Maximum Difference in Utility (B – A) | 0.7856 | |
| Minimum Difference in Utility (B – A) | -0.9535 | |

**Table 25. Comparison of two designs differing in gear ratio; the reference design (A) is fully specified**

| | Design A | Design B |
|---|---|---|
| Gear Ratio: | 2.1 | [1.9, 3] |
| Input Gear Diameter | 1.5 cm | 7 cm |
| Idler Gear Diameter | 1.5 cm | [1.5, 15] cm |
| Gear Width | 1.25 cm | [1.25, 8.75] cm |
| Gear Module | 6.35 mm/tooth | [1.27, 6.35] mm/tooth |
| | | |
| Maximum Utility | 0.950 | 0.866 |
| Minimum Utility | 0.733 | 0.00 |
| Maximum Difference in Utility (B – A) | -0.0390 | |
| Minimum Difference in Utility (B – A) | -0.9470 | |

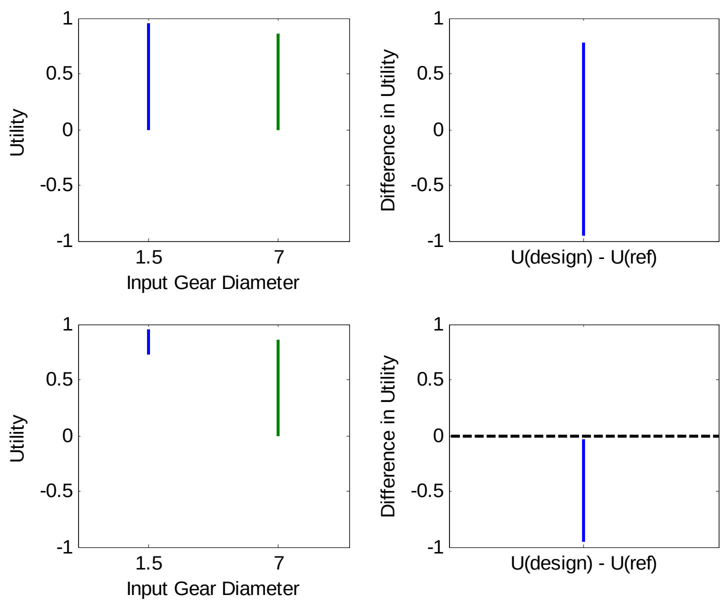



***Figure 83: Bounds of the two designs with respect to input gear diameter; bounds in the top plots are with none of the other design variables specified; bounds in the bottom plots are with a fully specified reference design. Elimination is only possible with a fully-specified reference design.***

While this demonstrates the importance of using a specific reference design for elimination, one should not mistake this to think that any reference design should be used. Instead, a good design is necessary, otherwise the tighter bounds can be worse than the design that one is trying to eliminate. This the choice of a good, specific reference design allows more elimination is possible because it takes advantage of shared uncertainty and removes the uncertainty about other design variables, as demonstrated in this section.

## *5.4 Summary of Example Design*

Through the example design in the previous sections, multiple aspects of the elimination method have been demonstrated, the advantages of the elimination method are more apparent, and the limitations of the method. In this section, this learning from the design example is reviewed.

### 5.4.1 Limitations

As pointed out in Section 0, there were some performance aspects of the elimination method that should be considered. The limitations with respect to those aspects are pointed out, as well some additional questions about performance that arose from the results of the example design. All of these limitations, some confirmed and some recognized from the design, are pointed out in this subsection.

Because of the uncertainty in design, significant elimination is not always possible. In the case of this design example, elimination is severely inhibited by the uncertainty encountered in some of the decisions. For the idler gear diameter, gear width, and gear module, no elimination was possible because of the uncertainty and the strong coupling of these design variables. Additionally, these design variables did not strongly affect design performance so the influence of uncertainty could not be overcome to eliminate significantly. Since this uncertainty and these decision are similar to those encountered in many design problems, the elimination method is limited in its usefulness for these problems.

While the design process can still converge if elimination is limited in a single decision, if elimination is limited enough in successive decisions then the ability to converge on the most preferred design is severely inhibited. The series of eliminations in the design example left a significant amount of the design space unspecified with a utility that could vary in utility by as much as 0.33 from the reference design. Significantly more elimination, based on assumptions about the uncertainty, was necessary to finalize the design. This demonstrates that while the elimination method may be useful for decreasing the number of design alternatives considered, it cannot finish the design process in many cases because of the uncertainty. Additional methods are necessary to finalize the design.

Additionally, the elimination method may not lead one to a final design that is any better than those obtained by point-and-iterate methods. In the example, the design obtained using the elimination was marginally better than the designs obtained through the point-and-iterate approaches. If the design is simple enough such that it has a continuous objective function, then the point and iterate method work just as well, and with less cost, than the elimination method. However, one the elimination-based B&B approach should be more successful in complicated designs where a point and iterate method could easily get caught in at a local maximum.

In applying the elimination method, one must characterize the uncertainty and compute the bounds based on that uncertainty. This can be an expensive process. In the design example, computing these bounds took approximately 100 times as long as computing the deterministic performance of the system, while this resulted in run times of no longer than a few seconds for the single variable decisions, in much larger design problems this computation time can be significant. More significant than this computation time, was the time necessary to set up the simulation to compute the bounds. Since most simulations do not incorporate uncertainty, the models must be created that can compute these bounds on system performance.

In addition to the problems already mentioned, representing the set of designs that result from the elimination is more difficult than typical designs. Product Data Management systems have been developed for representing actual design instances, but not sets of designs. Representing the set of designs in the example problem was not particularly difficult, but the definition of the set is difficult to visualize and relate, discouragingly this is just for a system of 5 design variables. In larger systems, the constraints that define the set may be excessive and difficult to deal with.

### 5.4.2 Advantages

The limitations presented in previous section may seem too imposing for one to develop this approach further; however, the advantages offered by this approach are numerous enough to merit one to consider the approach heavily in the future.

Foremost among these advantages is the fact that the method always includes the most-preferred design. This ensures that the design process progresses toward the most-preferred design and will always converge on a set that contains the most-preferred design. This allows the designers to disregard the designs that are eliminated, saving the time from ever investigating those designs again. It allows the designers to accurately assess, both by the set of designs remaining and the variation in utility, how close they are to the completed design. This contributes to a better organized and hopefully more efficient design process.

An additional contribution from this method is the realization that design performance should be compared based on their relative performance under the given uncertainty. This allows for more elimination based on the criterion that has been given, but considering shared uncertainty would result in better decision-making regardless of the decision criterion. One designing with probabilities or other uncertainty representations should consider this shared uncertainty as well. The underlying theory as to why this is a good idea was given in Section 4.3, while the advantages of this approach were demonstrated in this Chapter.

By considering the relative performance of the designs in decision-making, one also is taking advantage of another contribution in this work. By using a specified reference design, one decreases the uncertainty about other design variables. This allows more elimination regardless of the uncertainty that one is considering.

While the first advantage pointed out above shows the advantage of the particular method, the other two advantages obtained by considering relative performance are contributions abstracted from the method. Both the method and the contributions from the method are useful in design.

### 5.4.3 Role of this chapter in the thesis

In this chapter, the proposed elimination method was tested for usefulness in an example design of a SAE Mini-Baja Gearbox. It was found that the both the method and two components of the method were useful in engineering design. This result establishes the theoretical performance validity of the hypothesis and fulfills the two bottom squares on the validation square in Figure 84. With these two quadrants of the validation square completed, attention is turned to the final quadrant. The final quadrant is fulfilled and the results of thesis are reviewed in the next and final chapter. The role of this chapter in the thesis is shown in Table 26 and Figure 85.

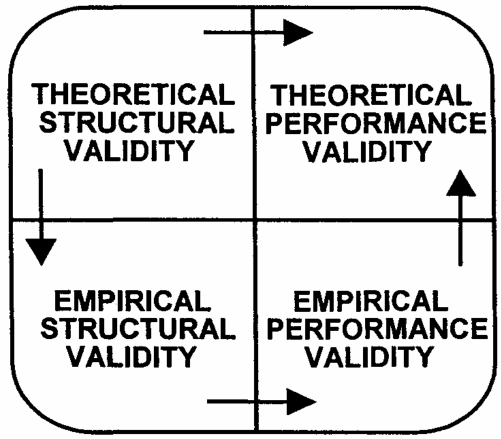



***Figure 84:*** ***Validation Square. The bottom two squares involve the usefulness of the method in engineering design; both were addressed in this chapter.***

**Table 26: Validation Strategy**

| **Quadrant of Validation Square** | **Thesis Chapter/Section** | **Aspect of Validation (for which Hypothesis)** |
|---|---|---|
| | Chapter 2 | Literature Search to establish theoretical basis for proposed method |
| Theoretical Structural Validity (1) | Chapter 4: Section 4.2 | Theoretical structural soundness of rational decision-making |
| | Chapter 4: Section 4.3 | Soundness of applying common uncertainty to elimination |
| **Empirical Structural Validity (2)** | **Chapter 5: Section 5.2** | **Example's capability to test elimination method is established** |
| **Empirical Performance Validity (3)** | **Chapter 5: Section 5.5** | **Example is evaluated to determine if design method is useful** |
| | **Chapter 5: Section 5.6** | **Example decisions examined to verify performance of elimination method** |
| Theoretical Performance Validity (4) | Chapter 6: Section 6.1 | Hypothesis is revisited and examined for validity to make 'Leap of Faith' |

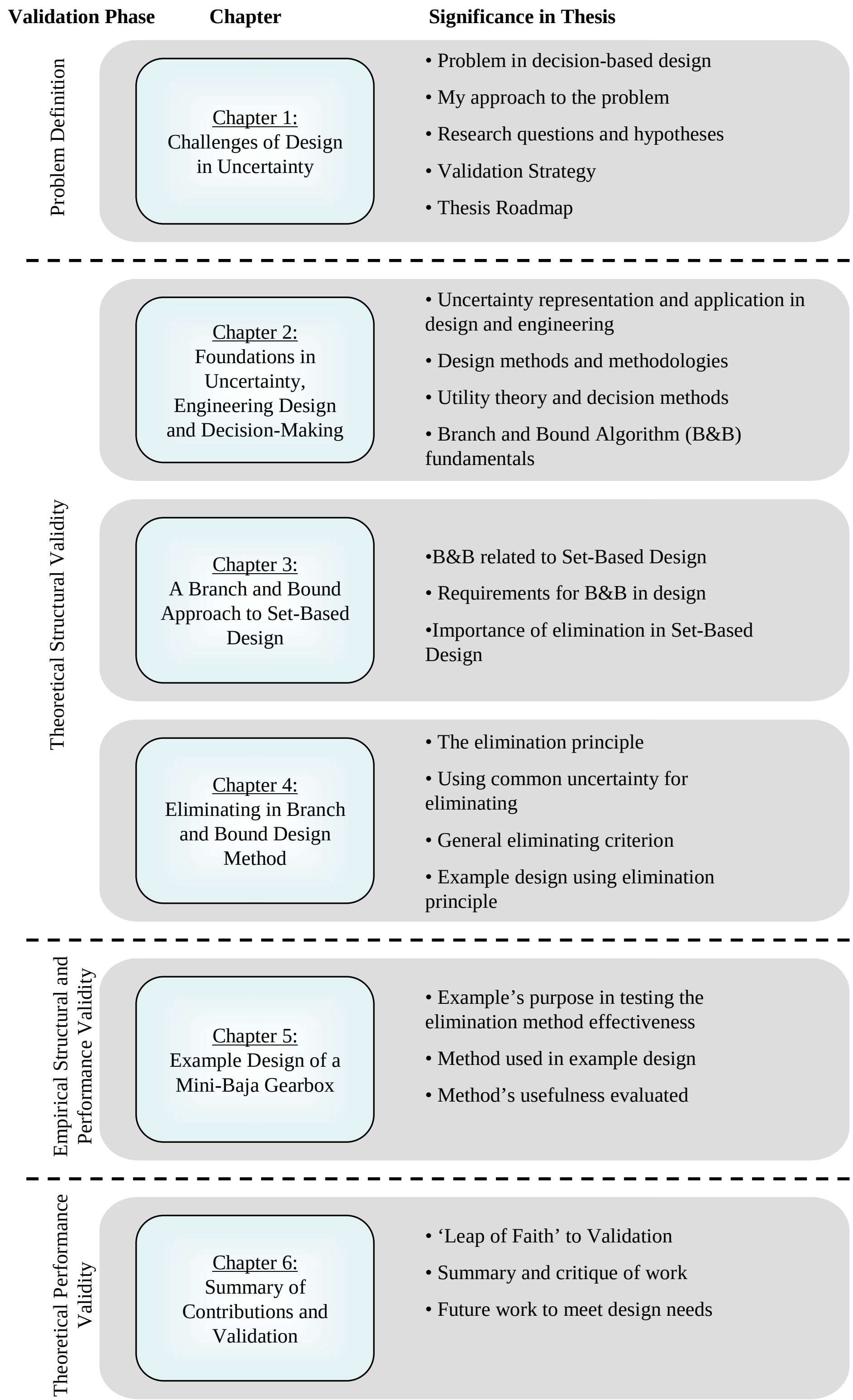



***Figure 85: Thesis Roadmap. With the conclusion of Chapter 6, the thesis is brought to a close.***

# CHAPTER 6

# SUMMARY AND ANALYSIS

In the previous chapters, a method for eliminating design alternatives under uncertainty was presented and validated in the context of a Branch and Bound approach to design. In this chapter, the validation process is brought to a close and their contributions from this work are analyzed for their usefulness and the opportunity for future advancement. In Section 6.1, the research question and hypothesis are revisited and then steps taken to validate the hypothesis are presented and explained in condensed form. Relying on a leap of faith, I then establish the theoretical structural validity. In Section 6.2, the work produced in this thesis is analyzed for the valuable contributions and limitations. Based on this analysis, I recognize the opportunities for future advancement in research in Section 6.3.

## *6.1 Validation of the Research Hypothesis*

In this thesis, I recognize that the design process contains numerous sources of uncertainty. Because of this uncertainty, finding the most-preferred design via a series of point selection is impossible, so instead I approach the design process from a different perspective. I still decouple the problem into a sequence of decisions, but because the most preferred solution cannot be determined via a series of point decisions, I do not force the decision maker to make point decisions. Instead, I propose that the designer decide on the *set* of design alternatives. In this *set-based design* approach, the designer decides on a set of possible solutions, eliminating the alternatives or values from the set that are guaranteed not to lead to the most preferred design based on and consistent with the currently available information and knowledge. Thus, the focus is on elimination rather than selection.

In this thesis, I address the issue of how one should eliminate. Specifically, the need was recognized for a method to eliminate design alternatives under epistemic uncertainty, and this is the basis of my research question:

> **Question:** *Under conditions of epistemic uncertainty, how should one eliminate design alternatives?*

The goal of the designer is to find the most preferred design, thus the designer should eliminate all designs that cannot be the most preferred design. This is rational elimination, and it serves as the underlying principle pointed out in my hypothesis:

> **Hypothesis:** *One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design to account for shared uncertainty.*

To validate this hypothesis, the validation square, shown in Figure 6.1 was used. This square guides one through the process of validating work in engineering design. In using the square, one establishes the theoretical structural validity of the hypothesis in the first quadrant (upper left) of the square. Then the lower two quadrants are used to establish that the performance validity, or usefulness, of the hypothesis in engineering design. In the last quadrant, one takes the leap of faith to assume the theoretical performance validity of the hypothesis. The first three quadrants of the square have been fulfilled in previous chapters and are explained briefly in Section 6.1.1. In section 6.1.2, I review the work performed in this thesis to take the leap of faith necessary for theoretical performance validity.

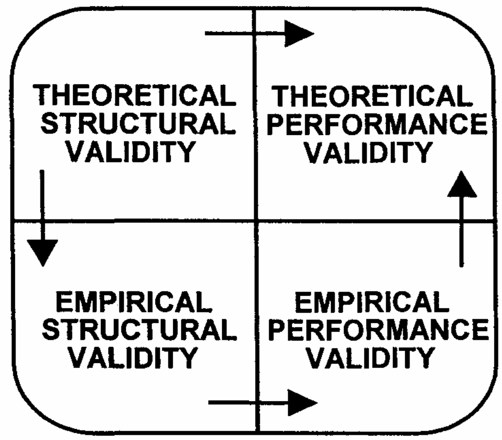



***Figure 86: Data of an uncertain parameter to be characterized with an uncertainty representation.***

### 6.1.1 Completed steps in Validating the Elimination Method

The first three steps in validating the elimination method have already been completed in the previous chapters. The specifics of these steps were explained in previous chapters and are summarized in this section.

**Theoretical Structural Validity**

As a first step in the process of validation, I check whether the proposed hypothesis is internally consistent and grounded in the foundation of previous work. In this section, I step through the theory that is applied to demonstrate this theoretical structural validity.

**Uncertainty Theoretical Foundation**

As pointed out in Chapter 1 and 2, uncertainty is ubiquitous in engineering design. First, systems are designed to perform in the future, which is uncertain. Second, design performance is predicted by using a model; either a prototype or a computational model. Because models are, by definition, an abstraction of reality, they cannot perfectly predict reality and their results contain some uncertainty. Lastly, the design process often is decomposed into a series of decisions and the designer addresses one decision at a time. (Sage 1977; Mistree, Smith et al. 1990; Hazelrigg 1998; Thurston 1999; Chen 2001; Chen 2003) This introduces uncertainty because in a sequence of decisions, the outcome of the future decisions, which affect performance, often are not known.

The term uncertainty is used to describe incomplete information. As pointed out in Chapter 2, researchers have recognized that this incomplete information arises from one of two basic phenomena: naturally random behavior, called aleatory uncertainty, or a lack of knowledge, called epistemic uncertainty (Antonsson and Otto 1995; Parry 1996; Oberkampf, DeLand et al. 2002). In this thesis, I have focused on epistemic uncertainty and I have chosen to represent that epistemic uncertainty with intervals. While this is an oversimplification of the problem, valuable insight can be gained by doing so, as explained in Chapter 2.

**Branch and Bound Theoretical Foundation**

The uncertainty in the design process makes finding the most preferred design difficult. As explained in Chapter 1, I approach this problem from the set-based design perspective. This approach builds in part on the Set-Based Concurrent Engineering (SBCE) approach at Toyota. Allen C. Ward documented this approach and recognized it as an important factor in the company's success (Ward, Liker et al. 1995; Sobek and Ward 1996), as pointed out in Chapter 2. Although the SBCE approach has proven successful at Toyota, it has not yet been formalized into a systematic method. In Chapter 3, I proposed such a method based design based on the Branch and Bound (B&B) Algorithm, which similar to the set-based design approach, search through a solution space using a process of elimination. The possible benefits of this approach also were pointed out in Chapter 3; however, these benefits only can be realized if the requirements of the method are met. Foremost among these requirements is the need for a method of elimination under uncertainty.

## Theoretical Foundation of Elimination

Since this approach is a systematic process of eliminating inferior design alternatives, a method for eliminating inferior design alternatives is needed. This elimination method is the basis of the research question and hypothesis:

> **Question:** *Under conditions of epistemic uncertainty, how should one eliminate design alternatives?*
>
> **Hypothesis:** *One should eliminate design alternatives rationally by comparing them to a detailed, specific reference design to account for shared uncertainty.*

The hypothesis has two main points. First, one should eliminate design alternatives rationally; this means that one needs to eliminate the design alternatives consistent with their preferences. The second aspect in the hypothesis relates to comparing the design alternatives to a specific reference design for elimination to incorporate shared uncertainty. Both of these aspects of the hypothesis were explained in Chapter 4 and are briefly here.

### Foundation of Rationality

As pointed out in Section 2.4.1, rational decision-making can be expressed mathematically in utility theory:

$$E[U(A)] > E[U(B)] \Leftrightarrow A \succ B$$

The most preferred alternative has the largest expected utility. While this accurately reflects one's preferences under uncertainty represented in probability distributions, it needs to be extended to account for intervals. Establishing dominance for interval-valued expected utilities requires comparison between intervals rather than scalars:

$$\forall x \in [\underline{X}, \overline{X}], \forall y \in [\underline{Y}, \overline{Y}] : x < y \Leftrightarrow \overline{X} < \underline{Y}$$

In section 4.2, this comparison is used to formulate the following criterion for elimination in set-based design. In this criterion, it is recognized that when there are more than two alternatives to be considered for elimination, one does not need to compare every alternative to every other. Instead, one alternative can be chosen as a reference. Since elimination can only occur if the lower bound of one design is larger than the upper bound of the design being eliminated, one should choose as a reference the alternative with the maximum lower-bound:

***Elimination Criterion***

*Consider the design space D. Consider design alternatives* $A_i \in D$ *with expected utilities* $U(A_i) \in [\underline{U}(A_i), \overline{U}(A_i)]$.
*One should eliminate* $A_i$ *if and only if* $A_i$ *is dominated by at least one other design:*

$$\exists A_j \in D : A_j \succ A_i$$

*or:*

$$\overline{U}(A_i) < \max_{A_j \in D} \underline{U}(A_j)$$

***Figure 87: Elimination criterion for interval uncertainty.***

Applying this elimination criterion results in a non-dominated solution set $S \subset D$ with:

$$\forall A_i, A_j \in S : A_i \nsucc A_j$$

As pointed out in Section 4.2, this has significant implications for the decision-making process: because of uncertainty, one may no longer be able to find a single design alternative that dominates all others (i.e., has the largest expected utility), but one may have to settle for a set of alternatives—the non-dominated set.

**Foundations of shared uncertainty**

While the elimination criterion given in Section 4.2 does eliminate dominated designs, uncertainty in the design process can limit the opportunity for elimination, as pointed out in that section. However, elimination can be improved by accounting for shared uncertainty, as explained in Section 4.3. Shared uncertainty is an uncertainty quantity that is the same value for multiple design alternatives. A similar concept is employed in Monte Carlo simulation in the form of Common Random Numbers and in Interval Analysis in the form of dependence, as pointed out in Section 2.1.

As explained in Section 4.3, the designer should account for shared uncertainty by considering the difference in performance for the same shared uncertain conditions. This approach is applied to expected utility as follows:

$$A \succ B \Leftrightarrow \forall z_s \in Z_s : U\left(A, z_s\right) - U\left(B, z_s\right) > 0$$

In Section 4.2, it was indicated that the largest number of design alternatives could be eliminated by using the alternative with the largest lower-bound on expected utility as a reference for comparison. However, in the case of shared uncertainty, one is concerned not just with the lower bound on reference design, but the designs performance throughout all the uncertain conditions. Therefore, it is no longer sufficient to use one single reference in elimination, as pointed out in Section 4.3.

To formulate the elimination criterion under shared uncertainty in the most general sense, one must distinguish between shared uncertainty, $z_s$, and uncertainty that is specific to each alternative, $z_i$ for alternative $A_i$. This leads to the final criterion for elimination under shared uncertainty:

***Elimination Criterion under Shared Uncertainty***

*Consider the design space D and design alternatives* $A_i \in D$ *with expected utilities* $U(A_i)$. *Consider the uncertainty* $z_s \in Z_s$ *shared by all alternatives in D and the uncertainties* $z_i \in Z_i$ *that are specific to each* $A_i \in D$.
*One should eliminate* $A_j$ *if and only if* $A_j$ *is dominated by at least one other design:*

$$\exists A_i \in D : A_i \succ A_j$$

*or:*

$$\max_{A_i \in D} \max_{\substack{z_c \in Z_c \\ z_j \in Z_j \\ z_i \in Z_i}} \left( U\left(A_j, z_j, z_c\right) - U\left(A_i, z_i, z_c\right) \right) < 0$$

***Figure 88: General elimination criterion that accounts for shared uncertainty with multiple reference designs***

In Section 4.3, it was pointed out that while this criterion is effective for eliminating fully specified design alternatives, the uncertainty about the specifics in a design alternative can severely limit one's ability to eliminate inferior design alternatives. I suggest an approach where one specifies the reference design completely in considering elimination. This reduces the uncertainty about the other design variables that are not considered and increases elimination.

This theoretical foundation is presented in Chapters 2-4, which establishes the theoretical structural validity of the hypothesis. With this first quadrant of the validation square completed, attention was turned to the quadrants two and three in Chapter 5 for empirical validation. A brief explanation of this empirical study is presented next.

**Empirical Structural and Performance Validity**

The usefulness of the elimination method was tested. This test was performed using the design of a gearbox for an SAE Mini-Baja car, and the details of this design are contained in Chapter 5; this process is explained briefly here.

As pointed out in Section 5.1.3, the gearbox design problem is a good test case for the proposed elimination method because it is a complex enough problem to test the method in an interesting experiment while it is simple enough that the method is not lost in the complexity of the problem. Thus, the example design problem met the Empirical Structural Validity.

With the validity of the design example established, attention was turned to the Empirical Performance Validity – how well the method performed in the example. This was addressed in Sections 5.2 and 5.3. Specifically, it was determined that the elimination method had mixed results for eliminating the designs under uncertainty. While elimination with respect to some design variables was significant, elimination with respect to other design variables was not always possible. As pointed out in Section 5.3, this shows the elimination method is effective for design variables that strongly effect performance but relatively ineffective for variables that weakly affect design performance and are strongly coupled. Most importantly, the method proved robust in not eliminating any of the design alternatives that could be the most-preferred.

Additionally, the method proved robust in converging on the most preferred design: the set that resulted from the series of eliminations contained the most preferred design for all of the uncertainty conditions investigated, as pointed out in Section 5.3.1. While the method converged on this set of design alternatives, the process was more costly than the typical point-and-iterate design approaches. However, the cost of the method could be improved significantly if there was computer software to support computing these bounds, as pointed out in Section 5.4. The additional cost that may result from the experiment is a reasonable price to pay for the robustness of the process, which always contains the most preferred design. Based on these results, it was determined that the elimination method is useful in the design process for eliminating design alternatives, thus significantly decreasing the feasible set.

### 6.1.2 Theoretical Performance Validity

While the previous chapters have shown that the method is theoretically sound and useful in the design process, this only covers the first three quadrants of the validation square. In this section, the fourth and final quadrant of the validation square is addressed: the Theoretical Performance Validity of the method.

In theoretical performance validation, one establishes not only is the method theoretically sounds and proved empirically sound, but that the method should theoretically perform well in the future. This validation requires multiple design examples with results that support the performance of the method. This is not possible in the case of the proposed method; however, the results from my investigation in this thesis look promising. I am confident in the performance of this method in future designs.

Since this method is theoretically sound and has proved capable of eliminating design alternatives in this typical engineering problem of reasonable scope, I believe that the method is capable of eliminating design alternatives in similar design problems. This fulfills the fourth and final quadrant in the validation square, as I have taken the ‘leap of faith’ to accepting the

Theoretical Performance Validity of my hypothesis. However, I make this projection with some limitation, which I explain in the next section.

## *6.2 Contribution and Critical Analysis of Contribution*

While the previous chapters have described and tested my research, in this section I critically review my research to extract the valuable contributions and limitations.

### 6.2.1 Summary of Contribution

Based on the research question and hypothesis, I see my contribution to the design community as follows:

**A method for eliminating design alternatives under interval uncertainty:** While researchers have developed means of decision-making under conditions where the uncertainty can be represented by a probability distribution (Luce and Raiffa 1957; Keeney and Raiffa 1993; Triantaphyllou 2000; Fernandez, Seepersad et al. 2001; Stirling 2003), a method for decision-making with intervals was needed. I believe that the proposed method meets this need.

**A means of reducing effect of shared uncertainty in the decision process:** Typically in the design process, there is environmental uncertainty that affects all the design alternatives being considered. I have recognized this uncertainty has dependence between the multiple design alternatives and I have determined how one can consider this *shared uncertainty* when eliminating designs. By comparing the designs based on relative performance, one is able to eliminate dependence and improve elimination. While considering this shared uncertainty was formulated for and demonstrated for interval uncertainty, there is no theoretical reason this process could not be extended to other uncertainty representations. Therefore, I believe this contribution can be applied to improve elimination and decision-making for other uncertainty representations.

**A means of reducing the effect of other unspecified design variables:** As pointed out in Chapter 1, 2, and 4, one often decides on some set of design variables while other design variables are not considered and not yet specified. This introduces considerable uncertainty in the design process that can limit elimination. To reduce the effect of this uncertainty, the designer needs to compare the designs to a reference design that is specified in detail. I believe that this approach is not only sound for interval uncertainty but also should work for other uncertainty representations.

These contributions have been validated in the context of engineering design. However, I have also introduced that concept of a Branch and Bound (B&B) approach to engineering design. This approach has not been formulated into a particular method, neither has it been tested in an engineering design problem. I pointed out the advantages of such an approach in Chapter 3 and the requirements that such an approach needed to fulfill. I believe that if this method could be realized, it may be more important than any of my contributions validated in this thesis. The first step in creating this approach was taken in this thesis, as I devised the means needed to eliminate designs under uncertainty.

### 6.2.2 Critique and Limitations of this contribution

While the contributions in this thesis are useful in engineering design, there are some significant limitations to this work. Namely, I see these limitations as follows:

**Elimination may be limited by the uncertainty in some design problems:** Because of the uncertainty in design, significant elimination is not always possible. In the case of this design example, elimination was inhibited by the uncertainty encountered in some of the decisions. Since this uncertainty and these decision are similar to those encountered in many design problems, the elimination method is limited in its usefulness for these problems. While this is the only rational elimination possible given the interval information, one may argue that the designer should assume more about the uncertainty to foster more elimination. While designers may assume this information, as was done to complete the example, this can lead one to an inferior design solution. Instead, I believe that elimination can be improved substantially by accounting for more detailed representations of uncertainty in the elimination criterion. The additional information contained in more detailed uncertainty representations should allow for more elimination.

**A design process based on this elimination may not converge toward a single design alternative:** While the design process can still converge if elimination is limited in a single decision, if elimination is limited enough in successive decisions then the ability to converge on the most preferred design is severely inhibited. The series of eliminations in the design example left a significant amount of the design space unspecified with a utility that could vary in utility by as much as 0.33 from the reference design. Significantly more elimination, based on assumptions about the uncertainty, was necessary to finalize the design. This demonstrates that while the elimination method may be useful for decreasing the number of design alternatives considered, it cannot finish the design process in many cases because of the uncertainty. Additional methods are necessary to finalize the design. However, the elimination method can be used successfully to narrow the set of designs considered, and I believe that more detailed representations of uncertainty can allow on to converge on a single design alternative

**In some designs the elimination method may not perform any better than the traditional point-and-iterate approaches:** The elimination method may not lead one to a final design that is any better than those obtained by point-and-iterate methods. In the example, the design obtained using the elimination was marginally better than the designs obtained through the point-and-iterate approaches. If the design is simple enough such that it has a continuous, monotonic objective function, then the point-and-iterate method may work just as well, and with less cost, than the elimination method. However, the elimination-based B&B approach should be more successful in complicated designs where a point and iterate method could easily get caught in at a local maximum. In this way, the designer pays an addition cost in using the elimination approach in return for the robustness offered by the approach.

**There could be significant cost in applying the method:** In applying the elimination method, one must characterize the uncertainty and compute the bounds based on that uncertainty. This can be an expensive process. In the design example, computing these bounds took approximately 100 times as long as computing the deterministic performance of the system, while this resulted in run times of no longer than a few seconds for the single variable decisions, in much larger design problems this computation time can be significant. More significant than this computation time, was the time necessary to set up the simulation to compute the bounds. Since most simulations do not incorporate uncertainty, the models must be created that can compute these bounds on system performance. As pointed out in Chapter 5, this could be overcome if simulation software computed these bounds.

**There could be significant cost is representing the feasible set:** In addition to the problems already mentioned, representing the set of designs that result from the elimination is more difficult than typical designs. Product Data Management systems have been developed for representing actual design instances, but not sets of designs. Representing the set of designs in the example problem was not particularly difficult, but the definition of the set is difficult to visualize and relate. Discouragingly this is just for a system of 5 design variables. In larger systems, the constraints that define the set may be excessive and difficult to deal with. However, this problem has not stopped Toyota from using a set-based approach, so there is a practical representation for these sets in the design process (Ward, Liker et al. 1995; Sobek and Ward 1996). Additionally, there has not been a pressing need to represent sets in research, and I believe that the description logic research community can sufficiently address this problem.

While these limitations seem daunting, I believe that future work can address these problems and extend the work in this thesis to be useful in engineering design at companies.

## *6.3 Future Work*

In this thesis, I developed a method for eliminating design alternatives under interval uncertainty as the first step in a formal approach to set-based design. In order to turn this into a complete, formal set-based design method, several additional issues need to be addressed.

Although a formal set-based design method seems more robust than traditional point design methods, the ultimate standard for comparison should be net value. Since the set-based design approach guarantees that the most preferred solution is never eliminated, one pays a price for achieving this. Since one must evaluate sets of design alternatives rather than point solutions, significant additional resources are required for solving the design problem. The question the is: Does this design approach result in better, more valuable design solutions even when taking into account the increased design cost? This issue will require further research, and the answer to the question will likely depend on the development of the method's components.

An elimination method for intervals was developed in this thesis, but there are other uncertainty representations that are applicable in the design process. The elimination method needs to be extended to include these valuable representations. For example, Augenbaugh and Paredis recognize imprecise probabilities to be a theoretically sound and valuable representation in engineering design (Aughenbaugh and Paredis 2005). The elimination method may be extended to this method by considering the intervals of the expected utility in the representation. Elimination for this representation needs to be investigated further.

For an efficient Branch and Bound design method, an efficient branching method needs to be formulated. In each branching step, a different axis of differentiation, or a different design attribute or characteristic is considered. The order in which these characteristics are considered potentially impacts the cost of the design process.  An efficient branching method probably should branch over a subset that allows the most elimination at a time. Developing such an efficient branching method will require considerable further research.

Another issue that requires further investigation is that of a stopping criterion. Since epistemic uncertainty can never be eliminated entirely, one will have to make a point decision ultimately to select a final design solution. At which level of uncertainty should one make this point decision? I anticipate that the answer will be determined based on economic considerations. At some point, additional investigation to reduce the epistemic uncertainty will be more expensive than the expected benefit in increased performance of the design solution. Although one could still "optimize" the performance of the product further, it is then better to stop the design process and use the best solution found so far. This issue has been recognized and relates to work in the value of information and hopefully can build on research in that area.

A final issue is the need for better representations and computational methods for set-based design. Interval representations and computational methods are still in the development stage and cannot yet be applied to complex engineering analyses such as computational fluid dynamics. In addition, once we move into the detailed design stage, one is limited to representing the geometry in terms of point solutions.

Once these issues have been addressed, the cost of implementing set-based design methods in a systematic manner could become economically viable. We hope that our colleagues will join us in the continued development of set-based design methods. The success of these methods and concepts will ultimately depend on their work.

# APPENDIX A

# MINI-BAJA DESIGN PROBLEM DETAILS

This appendix contains details from the design example problem given in Chapter 5. Specifically, the data and model for the torque-speed characteristic of the engine are given.

### Torque-Speed Characteristics of Mini-Baja Engine

The performance of a gearbox design is based on the acceleration and top speed of the overall car with that gearbox design, which depend strongly on the torque-speed characteristics of the engine. Different torque-speed characteristics may lead to a different gearbox design. Therefore, the torque-speed characteristics are given in this section of Appendix A.

All Mini-Baja cars use a Briggs & Stratton Model 20 engine in their competition car. According to the Briggs & Stratton product specifications for the Model 20, the engine has the horsepower-speed characteristics given in Figure 51 and the torque-speed characteristics given in Figure 90 (Briggs & Stratton 2004). These models for the engine performance would be correct if the engine was not modified based on the Society of Automotive Engineers' rules (Engineers) 2004). According to the rules of SAE Mini-Baja, the fuel to the engine is restricted to limit the speed of the engine to 3600 rpm instead of the standard 4000 rpm. While this may not seem like a significant change, because it is the fuel that is limited, it affects the engine characteristics throughout the range of speeds. As a result, the torque-speed characteristic of the engine is significantly different from the data given in Figure 51 and Figure 90. Instead the engine torque-speed curve was obtained from a dynometer and is given in Table 27. While these

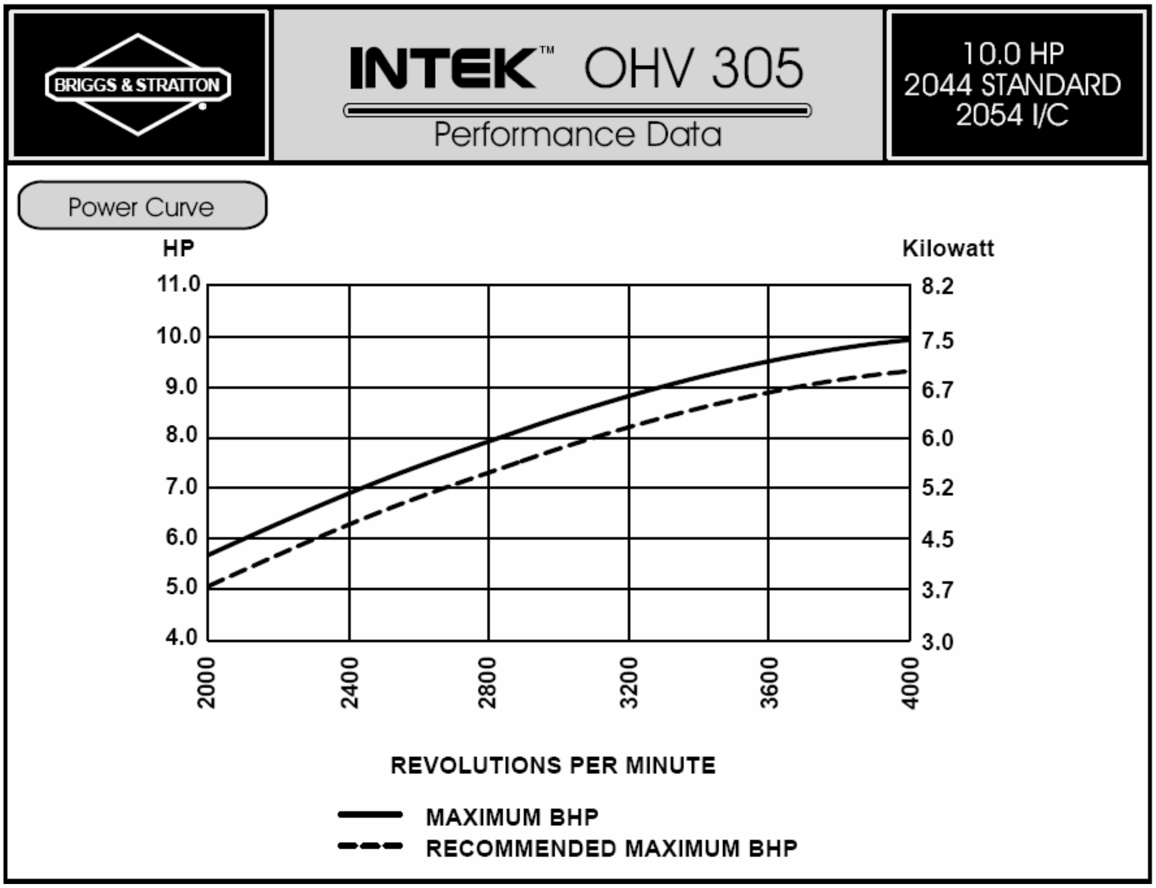



***Figure 89: The horsepower-speed characteristics of the Briggs & Stratton Model 20 engines.***

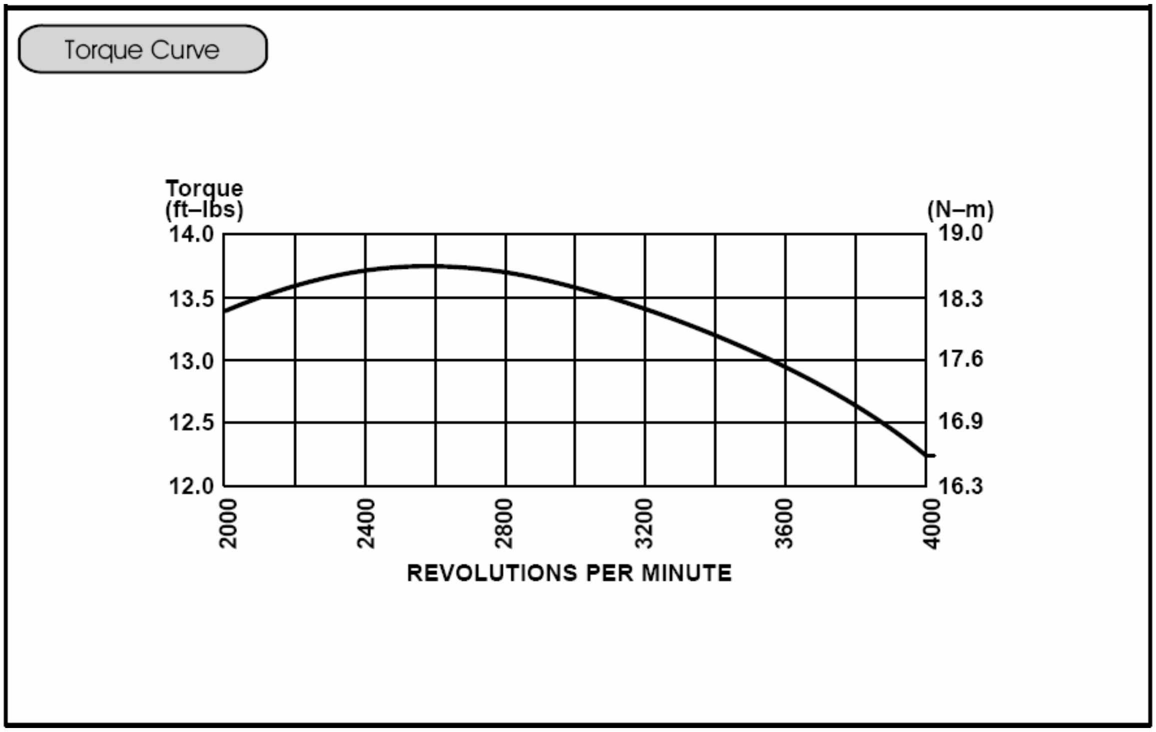



***Figure 90: The torque-speed characteristics of the Briggs & Stratton Model 20 engines.***

**Table 27. Engine torque curve characteristics obtained from a dynometer**

| Engine Speed (rpm) | Torque (n-m) |
|---|---|
| 0 | 4.00 |
| 400 | 6.00 |
| 800 | 7.00 |
| 1200 | 7.50 |
| 1600 | 7.00 |
| 2600 | 5.00 |
| 3600 | 0.00 |

results are accurate description of how the motor preformed on the dynometer, the results have to be made useful in the design process. To do this, they have to be modeled with the uncertainty of the model expressed. To do so, a third-order polynomial was fit to the torque-speed to obtain a model for the torque of the engine as a function of speed, and the resulting torque was bounded to include the uncertainty in the system. The data, the model, and the upper and lower bounds are presented in Figure 91. The results of this regression were as follows given in Figure 92.

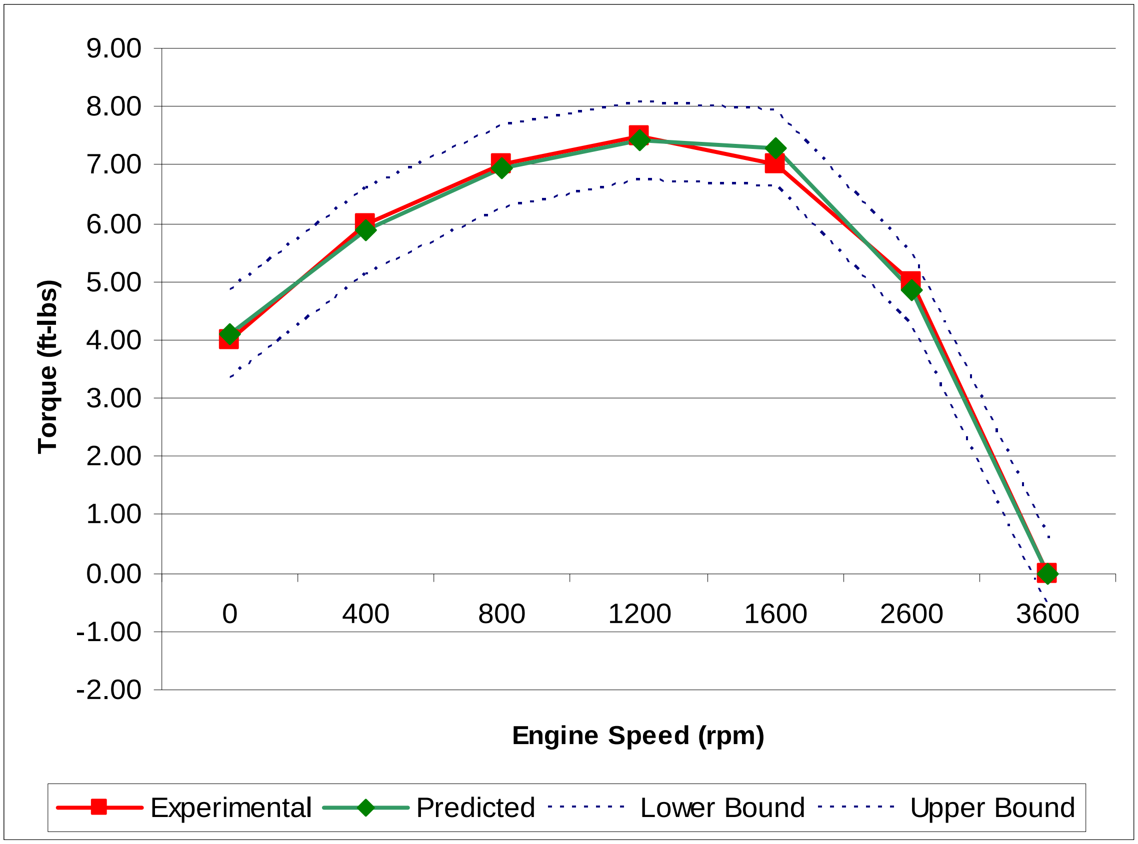



***Figure 91: The torque-speed characteristics of the Briggs & Stratton Model 20 engines: the experimental data from a dynometer, the third-order model, and the bounds on the results of that model.***

| Regression Statistics | |
|---|---|
| Multiple R | 0.998198439 |
| R Square | 0.996400124 |
| Adjusted R Square | 0.744600185 |
| Standard Error | 0.191923031 |
| Observations | 7 |

ANOVA

| | df | SS | MS | F | Significance F |
|---|---|---|---|---|---|
| Regression | 3 | 40.78123363 | 13.59374454 | 369.049752 | 0.00023829 |
| Residual | 4 | 0.147337799 | 0.03683445 | | |
| Total | 7 | 40.92857143 | | | |

| | Coefficients | Standard Error | t Stat | P-value | Lower 95% | Upper 95% | Lower 95.0% | Upper 95.0% |
|---|---|---|---|---|---|---|---|---|
| Intercept | 0 | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A | #N/A |
| X Variable 1 | -0.005765202 | 0.000428041 | -13.46882152 | 0.00017581 | -0.006953635 | -0.004576768 | -0.00695364 | -0.004576768 |
| X Variable 2 | -7.76757E-07 | 3.24173E-07 | -2.396114771 | 0.07467041 | -1.67681E-06 | 1.23295E-07 | -1.6768E-06 | 1.23295E-07 |
| X Variable 3 | 1.41268E-10 | 5.98672E-11 | 2.35969073 | 0.07768912 | -2.49503E-11 | 3.07486E-10 | -2.495E-11 | 3.07486E-10 |

***Figure 92: The regression analysis output.***

The model presented in Figure 91 was used to in the design process to bound the torque output of the engine. The equation for this model is as follows:

$$T_{\text{engine}}\left(\omega_{\text{engine}}\right)=B_1\left(\omega_{\max}-\omega_{\text{engine}}\right)+B_2\left(\omega_{\max}-\omega_{\text{engine}}\right)^2+B_3\left(\omega_{\max}-\omega_{\text{engine}}\right)^3\pm\varepsilon_{\text{torque}}$$

where $B_1$, $B_2$, and $B_3$ are given in Table 28. $\omega_{\max}$ is the maximum engine speed, 3600 rpm. $\varepsilon_{\text{torque}}$ is based on the following model:

$$\varepsilon_{\text{torque}}\left(\omega_{\text{engine}}\right)=A_1+A_2\omega_{\text{engine}}^2$$

where $A_1=0.66\,n*m$ and $A_2=3.1e-8\,n*m/rpm^2$. With these equations, the torque of the engine can be bounded for any given engine speed.

**Table 28. Parameter values for engine torque-speed model.**

| Parameter | Value |
|---|---|
| $B_1$ | -0.0058 n*m/(m/s) |
| $B_2$ | -7.8E-07 n*m/(m/s)$^2$ |
| $B_3$ | 1.4E-10 n*m/(m/s)$^3$ |

# REFERENCES


Adjiman, C. S., I. P. Androulakis and C. A. Floudas (1998). “A Global Optimization Method, Alpha-Bb, for General Twice Differentiable Npss: I - Theoretical Advances.” *Computers in Chemical Engineering*, **22**: 1137-1158.

Adjiman, C. S., I. P. Androulakis and C. A. Floudas (1998). “A Global Optimization Method, Alpha-Bb, for General Twice Differentiable Npss: Ii -Implementation and Computational Results.” *Computers in Chemical Engineering*, **22**: 1159-1179.

AGMA (1988). *Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth*. Alexandria, VA, American Gear Manufacturers Association.

AGMA (1989). *Geometry Factors for Determining the Pitting Resistance and Bending Strength of Spur, Helical, and Herringbone Gear Teeth*. Alexandria, VA, American Gear Manufacturers Association.

Alefeld, G. and J. Herberger (1983). *Introduction to Interval Computations*. New York, Academic Press.

Anscombe, F. J. and R. J. Aumann (1963). “A Definition of Subjective Probability.” *Annals of Mathematical Statistics*, **34**: 199-205.

Antonsson, E. K. and K. N. Otto (1995). “Imprecision in Engineering Design.” *Journal of Mechanical Design*, **117**: 25-32.

Appleyard, D. R. and A. J. Field Jr. (2001). *International Economics*. New York, McGraw-Hill.

Ashby, M. F. (1999). *Materials Selection in Mechanical Design*. Woburn, MA, Butterworth-Heinemann.

Aughenbaugh, J. M. (2004). "The Role and Limitations of Modeling and Simulation in System Design." *ASME International Mechanical Engineering Congress and RD&D Expo: Computers & Information in Engineering*, Anaheim, California, November 13-19, 2004.

Aughenbaugh, J. M. and C. J. J. Paredis (2005). "The Value of Using Imprecise Probabilities in Engineering Design." *Design Engineering Technical Conferences and Computers and Information Engineering*, Long Beach, CA, ASME.

Balakrishan, V., S. Boyd and S. Balemi (1991). “Branch and Bound Algorithm for Computing the Minimum Stability Degree of Parameter-Dependent Linear Systems.” *International Journal of Robust and Nonlinear Control*, **1**(4): 295-317.

Bayes, T. (1763). “An Essay Towards Solving a Problem in the Doctrine of Chances.” *Philosophical Transactions of the Royal Society of London*, **53**: 370-418.

Belegundu, A. D. and T. R. Chandrupatla (1999). *Optimization Concepts and Applications in Engineering*. Upper Saddle River, New Jersey, Prentice-Hall, Inc.

Ben-Haim, Y. (2001). *Information Gap Decision Theory: Decision under Severe Uncertainty*. Bodmin, Cornwall, Great Britain, MPG Books Ltd.

Berleant, D., M. Cheong, C. Chu, Y. Guan, A. Kamal, G. Sheble, S. Ferson and J. F. Peters (2003). “Dependable Handling of Uncertainty.” *Reliable Computing*, **9**(6): 407-418.

Bernoulli, D. (1738). "Exposition of a New Theory on the Measurement of Risk." *Econometrica*. **22:** 23-26.

Bernoulli, J. (1713). "Ars Conjectandi." *Basel*.

Boyd, S., A. Ghosh and A. Magnani (2003). "Branch and Bound Methods." Stanford University, Palo Alto, CA.

Briggs & Stratton, M. (2004). "Racing Engines: Model 20 Engine." (accessed November 30, 2004).

Broadwater, R. P., H. E. Shaalan, W. J. Fabrycky and R. E. Lee (1994). “Decision Evaluation with Interval Mathematics: A Power Distribution System Case Study.” *IEEE Transaction on Power Delivery*, **9**(1): 59-67.

Buede, D. M. (2000). *The Engineering Design of Systems*. New York, John Wiley & Sons, Inc.

Chang, T.-S. (1994). “Conceptual Robustness in Simultaneous Engineering: A Formulation of Continuous Spaces.” *Research in Engineering Design*.

Chapra, S. C. and R. P. Canale (1998). *Numerical Methods for Engineers: With Programming and Software Applications*. Boston, WCB / McGraw-Hill.

Chen, W. (2001). "An Approach to Decision-Based Design." *ASME Design Engineering Technical Conferences*, Pittsburgh, Pennsylvania, September 9-12, 2001.

Chen, W. (2003). “An Approach to Decision-Based Design with Discrete Choice Analysis for Demand Modeling.” *ASME Journal of Mechanical Design*, **125**: 490-497.

Chen, W., J. K. Allen, D. N. Marvis and F. Mistree (1996). “A Concept Exploration Method for Determining Robust Top-Level Specifications.” *Engineering Optimization*, **26**: 137-158. http://scholar.google.com/url?q=http://ideal.mech.northwestern.edu/pdf/EO96.ps.

Chen, W., J. K. Allen, K.-L. Tsui and F. Mistree (1996). “A Procedure for Robust Design: Minimizing Variations Caused by Noise Factors and Control Factors.” *ASME Journal of Mechanical Design*, **118**: 478-485.

Clausen, J. (1997). “Branch and Bound Algorithms - Principles and Examples.” *Parallel Computing in Optimization*: 239-267.

Clausen, J. and M. Perregaard (1997). “Solving Large Quadratic Assignment Problems in Parallel.” *Computational Optimization and Applications*, **8**(2): 111-127.

Clausen, J. and M. Perregaard (1999). “On the Best Strategy in Parallel Branch-and-Bound - Best-First-Search Vs. Lazy Depth-First-Search.” *Annals of Operations Research*, **90**: 1-17.

Clausen, J. and J. L. Traff (1991). “Implementation of Parallel Branch-and-Bound Algorithms - Experiences with the Graph Partitioning Problem.” *Annals of Operations Research*, **33**: 331 - 349.

Costa, R. and D. K. Sobek II (2003). "Iteration in Engineering Design: Inherent and Unavoidable or Product of Choices Made?" *DETC: Computers and Information in Engineering*, Chicago, Illinois, USA, September 2-6, 2003, ASME.

Creedy, J. (1986). *Edgeworth and the Development of Neoclassical Economics*. Oxford, Basil Blackwell.

Davis, L. (1991). *Handbook of Genetic Algorithms*. New York, Van Nostrand Reinhold.

de Moivre, A. (1756). *Doctrine of Chances*. London, A. Millar.

de Weck, O. L. (2002). "Isoperformance: An Alternative Design Methodology for Engineering Systems." Massachusetts Institute of Technology, Engineering Systems Division.

Devore, J. L. (2000). *Probability and Statistics*. Pacific Grove, CA, Duxbury.

Edgeworth, F. Y. (1881). *Mathematical Psychics*. London, Kegan Paul & Co.

Elishakoff, I. (2004). *Safety Factors and Reliability: Friends or Foes?* Dordrecht, The Netherlands, Kluwer Academic Publishers.

Engineers), S. S. o. A. (2004). "Sae Collegiate Design Series: Mini Baja." (accessed December 5, 2004).

Fernandez, M. G., C. C. Seepersad, D. Rosen, J. K. Allen and F. Mistree (2001). "Utility-Based Decision Support for Selection in Engineering Design." *Design Engineering Technical Conference*, Pittsburgh, Pennsylvania, September 9-12, 2001, ASME.

Ferson, S. and L. R. Ginzburg (1996). “Different Methods Are Needed to Propagate Ignorance and Variability.” *Reliability Engineering and Safety Systems*, **54**: 133-144.

Finch, W. W. (1997). *Predicate Logic Representations for Design Constraints on Uncertainty Supporting the Set-Based Design Paradigm*. Thesis, The University of Michigan. Ann Arbor. http://web.mit.edu/wfinch/www/research/umi_contact.html wfinch@mit.edu.

Finch, W. W. (1997). "A Set-Based System for Eliminating Infeasible Designs in Engineering Problems Dominated by Uncertainty." *Design Engineering Technical Conferences*, Sacramento, California, September 14-17, ASME. http://web.mit.edu/wfinch/www/research/publications/DTM97.pdf

http://web.mit.edu/wfinch/www/research/index.html.

Fishman, G. S. (2001). *Discrete-Event Simulation: Modeling, Programming, and Analysis*. New York, Springer-Verlag.

Ford, D. N. and D. K. Sobek II (2004). “Explaining the Second Toyota Paradox by Modeling Real Options in Product Development.” *IEEE Transaction on Engineering Management*.

Forsberg, K. and H. Mooz (1991). "The Relationship of Systems Engineering to the Project Cycle." *Proceedings of the National Council for System Engineering Symposium*, Chattanooga, TN, October 1991, 57-65.

Forsberg, K. and H. Mooz (1992). “The Relationship of Systems Engineering to the Project Cycle.” *Engineering Management Journal*, **4**(3): 36-43.

Forsberg, K. and H. Mooz (1996). *Visualizing Project Management*. New York, John Wiley & Sons, Inc.

Glover, F. (1986). “Future Paths for Integer Programming and Links to Artificial Intellegence.” *Computers and Operations Research*, **5**: 533-549.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York, Addison-Wesley.

Gu, X., J. E. Renaud, L. M. Ashe, S. M. Batill and A. S. Budhiraja (2000). "Decision-Based Collaborative Optimization under Uncertainty." *ASME Design Technical Conference*, Baltimore, MD, September 2000, DETC2000/DAC-14297.

Hansen, E. and G. W. Walster (2004). *Global Optimization Using Interval Analysis*. New York, Marcel Dekker, Inc.

Hayes, B. (2003). “A Lucid Interval.” *American Scientist*, **91**(6): 484-488.

Hazelrigg, G. A. (1996). *Systems Engineering: An Approach to Information-Based Design*. Upper Saddle River, NJ, Prentice Hall.

Hazelrigg, G. A. (1998). “A Framework for Decision-Based Engineering Design.” *ASME Journal of Mechanical Design*, **120**(4): 653-658.

Hoffman, D. R. (1998). "Overview of Concurrent Engineering." *Reliability and Maintainability Symposium*, Anaheim, CA, Jan 19-22, 1998, IEEE, Piscataway, NJ, 1-7.

http://ippserv.rug.ac.be/ (2005). "The Imprecise Probabilities Project." (accessed May 2005).

Karasan, O. E., M. C. Pinar and H. Yaman (2002). “The Robust Shortest Path Problem with Interval Data.” *Computers and Operations Research*.

Kearfott, R. B. (1997). “Empirical Evaluation of Innovations in Interval Branch and Bound Algorithms for Nonlinear Systems.” *Journal of Scientific Computing*, **18**: 574.

Kearfott, R. B. and V. Kreinovich (1996). *Applications of Interval Computations*. AH Dordrecht, The Netherlands, Kluwer Academic Publishers.

Keeney, R. L. (1974). “Multiplicative Utility Functions.” *Operations Research*, **1**: 23-25.

Keeney, R. L. and H. Raiffa (1993). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York, NY, Cambridge University Press.

Koyluoglu, H. U., A. S. Cakmak and S. R. K. Nielsen (1995). “Interval Algebra to Deal with Pattern Loading and Structural Uncertainties.” *Journal of Engineering Mechanics*, **121**(11): 1149-1275.

Laursen, P. S. (1993). “Simple Approaches to Parallel Branch and Bound.” *Parallel Computing*, **19**: 143-152.

Law, A. M. and W. D. Kelton (2000). *Simulation Modeling and Analysis*. Boston, MA, McGraw-Hill Companies, Inc.

Lawler, E. L. and D. E. Wood (1966). “Branch and Bound Methods: A Survey.” *Operations Research*, **14**: 699-719.

Liker, J. K., D. K. Sobek II, A. C. Ward and J. J. Cristiano (1996). “Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering.” *IEEE Transaction on Engineering Management*, **43**(2): 165-178.

Ling, J. and M. Brums (2004). "Me 6103 Semester Project." Class Report, Georgia Institute of Technology, Atlanta, GA**:** 56.

Luce, R. D. and H. Raiffa (1957). *Games and Decisions*. New York, John Wiley.

Malak Jr., R. J. and C. J. J. Paredis (2004). "On Characterizing and Assessing the Validity of Behavioral Models and Their Predictions." *2004 ASME DETC*.

Martin Sprocket & Gear, I. (2003). "U.S. List Price Sheet." Catalog, Martin Sprocket & Gear, Inc., Arlington, TX**:** 60.

Mistree, F., W. F. Smith, B. A. Bras, J. K. Allen and D. Muster (1990). "Decision-Based Design: A Contemporary Paradigm for Ship Design." *Society of naval architects and marine engineers annual meeting*, San Francisco, CA, October 31 - November 3, 1990, Society of naval architects and marine engineers. http://www.srl.gatech.edu/library/corepapers/MISTREE_12_90_SNAME_TRANS.pdf.

Montemanni, R. and L. M. Gambardella (2002). "An Algorithm for the Relative Robust Shortest Path Problem with Interval Data." Technical Report, IDSIA.

Montemanni, R., L. M. Gambardella and A. V. Donati (2004). “A Branch and Bound Algorithm for the Robust Shortest Path Problem with Interval Data.” *Operations Research Letters*, **32**: 225-232.

Moore, R. E. (1966). *Interval Analysis*. Englewood Cliffs, NJ, Prentice-Hall, Inc.

Moore, R. E. (1979). *Methods and Applications of Interval Analysis*. Philadelphia, Society for Industrial and Applied Mathematics.

Moore, R. E. (1991). “Global Optimization to Prescribed Accuracy.” *Computers and Mathematics with Applications*, **21**(6/7): 25-39.

Munson, B. R. (2001). *Fundamentals of Fluid Mechanics, 4th Edition*, John Wiley & Sons Inc.

Nemhauser, G. L., A. H. G. Rinnooy Kan and M. J. Todd (1989). *Optimization*. New York, Elsevier Science Publishers B.V.

Neyman, J. (1937). *Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability*. London, Royal Society.

Neyman, J. (1977). “Frequentist Probability and Frequentist Statistics.” *Synthese*, **36**(1): 97-131.

Nikolaidis, E., D. M. Ghiocel and S. Singhal (2005). *Engineering Design Reliability Handbook*. Boca Raton, Florida, CRC Press.

Norkin, V. I., G. C. Pflug and A. Ruszczynski (1996). "A Branch and Bound Method for Stochastic Global Optimization." *International Institute for Applied Systems Analysis*, Laxenburg, Austria.

Norton, R. L. (2000). *Machine Design: An Integrated Approach*. Upper Saddle River, NJ, Prentice-Hall.

Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert and K. F. Alvin (2002). “Error and Uncertainty in Modeling and Simulation.” *Reliability Engineering and Safety Systems*, **75**(3): 333-357.

Pahl, G. and W. Beitz (1996). *Engineering Design: A Systematic Approach*. New York, Springer-Verlag London Limited.

Parry, G. W. (1996). “The Characterization of Uncertainty in Probabilistic Risk Assessment of Complex Systems.” *Reliability Engineering and Safety Systems*, **54**(2): 119-126.

Parunak, H. V. D., A. Ward, M. Fleisher and J. Sauter (1997). "A Marketplace of Design Agents for Distributed Concurrent Set-Based Design." *ISPE International Conference on Concurrent Engineering: Research and Applications (ISPE/CE97)*, Troy, Michigan, August 20-22, 1997, ISPE.

Pederson, K., J. Emblemsvag, R. Bailey, J. K. Allen and F. Mistree (2000). "Validating Engineering Design Research: The Validation Square." *ASME Design Engineering Technical Conferences*, Baltimore, MD, September 10-14, 2000, ASME.

Rekuc, S. J. and C. J. J. Paredis (2005). "Considering Shared Epistemic Uncertainty in Set-Based Design." *submitted to: Design Engineering Technical Conferences & Computers and Information in Engineering*, Long Beach, CA, USA, September 24-28, 2005, ASME, DETC2005 - 85344.

Riley, W. F., L. D. Sturges and D. H. Morris (1995). *Statics and Mechanics of Materials: An Integrated Approach*. New York, John Wiley & Sons, Inc.

Ross, S. M. (1997). *Introduction to Probability Models*. San Diego, CA, Academic Press.

Sage, A. P. (1977). *Methodology for Large-Scale Systems*. New York, McGraw-Hill Companies.

Schmeidler, D. (1989). "Subjective Probability and Expected Utility without Additivity." *Econometrica*, **57**(3): 571-587.

Seepersad, C. C. (2001). *The Utility-Based Compromise Decision Support Problem with Applications in Product Platform Design*. Thesis. G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology. Atlanta, GA.

Simon, H. A. (1982). *Models of Bounded Rationality*. Cambridge, MIT Press.

Simon, H. A. (1996). *The Sciences of the Artificial*. Cambridge, Massachusetts, The MIT Press.

Simpson, T. W., D. Rosen, J. K. Allen and F. Mistree (1997). "Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design."

Sobek, D. K. I. and A. C. Ward (1996). "Principles from Toyota's Set-Based Concurrent Engineering Process." *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, California, August 18-22.

Sobek II, D. K. (2004). "Explaining the Second Toyota Paradox by Modeling Real Options in Product Development." *IEEE Transaction on Engineering Management*.

Stirling, W. C. (2003). *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge, United Kingdom, Cambridge University Press.

Taguchi, G. (1987). *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. White Plains, N.Y., UNIPUB/Kraus International Publications.

Thurston, D. L. (1999). "Real and Perceived Limitations to Decision Based Design." *ASME Design Technical Conference*, Las Vegas, NV, September 1999, ASME, DETC99/DTM-8750.

Triantaphyllou, E. (2000). *Multi-Criteria Decision Making Methods: A Comparative Study*. Dordrecht, The Netherlands, Kluwer Academic Publishers.

Tribus, M. (1969). *Rational Descriptions, Decisions and Designs*. Elmsford, NY, Pergamon Press.

von Neumann, J. and O. Morgenstern (1944). *Theory of Games and Economic Behavior*. Princeton, NJ, Princeton University Press.

Walley, P. (1991). *Statistical Reasoning with Imprecise Probabilities*. London, Chapman and Hall.

Ward, A. C., J. K. Liker, J. J. Cristiano and D. K. Sobek II (1995). “The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster.” *Sloan Management Review*, Spring 1995: 43-61

Winkler, R. L. (1996). “Uncertainty in Probabilistic Risk Assessment.” *Reliability Engineering and Safety Systems*, **54**(2-3): 127-132.

Wojtkiewicz, S. F., M. S. Eldred, R. V. Field, A. Urbina and J. R. Red-Horse (2001). “Uncertainty Quantification in Large Computational Engineering Models.” *American Institute of Aeronautics and Astronautics*, **14**.

Wu, S. (1999). “A Probabilistic Model of Set-Based Design.” *MIT Journal of Undergraduate Mathematics*, **1**(1): 189-195. http://www-math.mit.edu/phase2/UJM/vol1/WU-F.PDF.

Yaman, H., O. E. Karasan and M. C. Pinar (2001). “The Robust Spanning Tree Problem with Interval Data.” *Operations Research Letters*, **29**: 31-40.